Design of Morlet Wavelet Artificial Neural Network for Solving Two-Species Competition Model

By

Nimra Noor

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

November, 2025

Design of Morlet Wavelet Artificial Neural Network for Solving Two-Species Competition Model

By

Nimra Noor

MS-Math, National University of Modern Languages, Islamabad, 2025

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In **Mathematics**

To

FACULTY OF ENGINEERING & COMPUTING

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

NATIONAL UNIUVERSITY OF MODERN LANGUAGES
FACULTY OF ENGINEERING AND COMPUTING

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computing for acceptance.

Thesis Title: Design of Morlet Wavelet Artificial Neural Network for Solving Two-Species Competition Model

Submitted By: Nimra Noor **Registration #:** 92/MS/MATH/F23

Master of Science in Mathematics

Title of the Degree

Mathematics

Name of Discipline

Dr. Muhammad Rizwan

Name of Research Supervisor

Signature of Research Supervisor

Dr. Farhad Muhammad Riaz

Name of Research Co-Supervisor

Signature of Research Co-Supervisor

Dr. Anum Naseem

Name of HOD

Signature of HOD

Dr. Noman Malik

Name of Dean (FEC)

Signature of Dean (FEC)

November, 2025

AUTHOR'S DECLARATION

I Nimra Noor

Daughter of Raja Mazhar Hussain

Discipline Mathematics

Candidate of Master of Science in Mathematics at the National University of Modern Languages do hereby declare that the thesis Design of Morlet Wavelet Artificial Neural Network for solving Two-Species Competition Model submitted by me in partial fulfillment of MS degree, is my original work and has not been submitted or published earlier. I also solemnly declare that it shall not, in the future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be canceled and the degree revoked.

Signature of Candidate

Nimra Noor

Name of Candidate

25 November, 2025

Date

ABSTRACT

Title: Design of Morlet Wavelet Artificial Neural Network for Solving Two-Species Competition Model

Artificial Neural Networks (ANNs) have gained significant interest in solving mathematical and biological problems due to their powerful learning and approximation capabilities. The study of Two-Species Competition Model using nonlinear differential equations is a crucial area of computational mathematics and biomathematics. This thesis introduces a novel computational approach using ANN and a hybrid optimization framework to study the dynamics of species interactions. This thesis developed a hybrid optimization method using Sequential Quadratic Programming (SQP) and Genetic Algorithm (GA) to precisely approximate the solution of the Two-Species Competition Model. This model captures competitive behavior between two biological species over time using a feedforward ANN architecture with a Morlet wavelet (MW) activation function for enhanced learning capacity. The ANN-GA-SQP method is designed to efficiently and accurately solve complex ecological systems, demonstrating its potential as a powerful tool for modeling and understanding complex ecosystems. To verify the robustness, accuracy, and consistency of the proposed ANN-based approach 50 experimental runs were conducted for each test scenario of the Two-Species Competition Model. The hybrid GA-SQP optimized ANN model outperforms conventional numerical methods and hybrid optimization techniques in terms of convergence dependability, numerical stability, and predictive accuracy, as evaluated using statistical measures like Mean Absolute Deviation (MAD) and Mean Square Error MSE analysis. Overall the study demonstrates the effectiveness of neuroevolutionary methods in solving nonlinear differential equations in ecological modeling. The GA-SQP optimized ANN framework, incorporating Morlet wavelet activation function, offers a reliable, adaptable, and biologically inspired computational tool for complex dynamical systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
ANN	-	Artificial Neural Network
NN	-	Neural Network
FFNN	-	Feed Forward Neural Network
DE	-	Differential Equation
PINN	-	Physics Informed Neural Network
MW	-	Morlet wavelet
ODE	-	Ordinary Differential Equation
PDE	-	Partial Differential Equation
FDE	-	Fractional Differential Equation
RK	-	Runge-Kutta
ASA	-	Active Set Algorithm
W-fluid	-	Williamson Fluid
PPM	-	Predator Prey Model
WNN	-	Wavelet Neural Network
IPA	-	Interior Point Algorithm
LADM	-	Laplace Adomian Decomposition Method
DAE	-	Differential Algebraic Equation
DDE	-	Delay Differential Equation
SDE	-	Stochastic Differential Equations
IVP	-	Initial Value Problem
FDM	-	Finite Difference Method
INNS	-	International Neural Network Society

ACKNOWLEDGMENT

First and foremost, I would want to thank Almighty Allah from the bottom of my heart for making this study successful. I would like to sincerely thank my Supervisor Dr. Muhammad Rizwan and Co-Supervisor Dr Farhad Muhammad Raiz for their encouragement and support, without them this work would not have been possible. My sincere gratitude is also extended to the faculty and staff of Mathematics Department, who have given me the resources I needed to conduct my study as well as a helpful academic atmosphere.

DEDICATION

"This thesis work is dedicated to my parents, family, and teachers throughout my educational career who have not only taught me to work hard for the things I aspire to achieve but also who admired me unconditionally".

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

This chapter provides an overview of various relevant studies and papers within this specific field of study. This also highlights numerous relevant papers and literature related to the research topic that enhance the fundamental understanding of the subject through a numerical simulation process.

Artificial neural networks (ANN) are increasingly used to solve differential equations, offering an alternative to traditional numerical methods for example the Euler Method, Taylor series, etc. Various studies have proposed ANN architecture for solving different types of differential equations for example PDE, ODE etc. Zulqurnain Sabir *et al.* [1] represented the numerical solutions of the nonlinear mathematical model of Leptospirosis disease (LD) through ANNs for enhanced computational efficiency, along with optimization methodologies like genetic algorithm (GA) and ASA framework, referred to as ANNs-GA-ASA. Asif Zahoor *et al.* [2] proposed a study on the use of an artificial neural network and Bayes-Reg technique to improve the reliability, efficiency, and accuracy of dynamic calculations in HIV infection models. Muhammad Asif Zahoor Raja *et al.* examined the behavior of the W-fluid under stretching flow using a neural network backpropagation technique, demonstrating its robust efficacy [3]. Muhammad Asif Zahoor Raja *et al.* [4] introduced ANN-GA-SQP, a novel method for solving prediction differential models, combining genetic algorithms, sequential quadratic programming, and ANN for accuracy, efficiency, and reliability.

Sivalingam S M *et al.* [5] proposed an approach for finding solutions for FDE using a PINN, employing a constrained expression trial solution and an average and subtraction-based optimizer algorithm. Muhammad Asif Zahoor Raja *et al.* [6] developed a new methodology for handling PDE with fractional derivatives using ANN, utilizing the Hermite wavelet neural network framework for numerical solutions. Muhammad Kashan Basit *et al.* [7] discussed the use of ANN and SQP to solve second-order differential equations, enhancing results with log-sigmoid. Li Yan *et al.* [8] discussed the use of artificial neural networks, specifically the GA and SQP scheme, to address the nonlinear Liénard model. Asif Zahoor Raja *et al.* [9] presented a MW-NNs technique to obtain solution of the PPM, using global optimization and local optimization algorithms, confirming its accuracy and reliability. Chetna Biswas *et al.* [10] developed a neural network technique to solve a fractional order nonlinear reaction-advection-diffusion equation using Shifted Legendre orthogonal polynomials, utilizing fractional-order derivative characteristics for loss function calculation. Mingqiu Wu *et al.* [11] presented a WNN to solve FDE, using wavelet functions and a 1xNx1 structure, with simulation outcomes demonstrating its effectiveness. Zulqurnain Sabir *et al.* [12] presented a numerical solution for a nonlinear functional differential equation using the Functional Mayer Artificial Neural Network (FM-ANN), GA, and SQP. Korhan Günel *et al.* [13] explored feed-forward neural networks for solving differential equations with Dirichlet boundary conditions using swarm intelligence techniques, comparing their efficiency with traditional methods. Zulqurnain Sabir *et al.* [14] presented an ANN technique for solving multi-pantograph delay differential equations, demonstrating its accuracy and effectiveness through successful resolution of three second-order MP-DDE problems. Zulqurnain Sabir *et al.* [15] introduced a novel computational approach, GNNs-GA-SQP, to numerically address singular periodic nonlinear differential systems in nuclear physics, evaluating its effectiveness through two SP-NDS problems. Neha Yadav *et al.* [16] introduced the harmony search algorithm (HSA) and artificial neural networks (ANN) for numerically solving differential equations, reducing error and producing approximation results Zulqurnain Sabir *et al.* [17] used Artificial neural networks and Levenberg-Marquardt backpropagation technique to minimize mean squared error in a nonlinear dengue fever SIR system, assessing its performance, accuracy, dependability, and efficacy. Zulqurnain Sabir *et al.* [18] addressed an Emden Fowler system using artificial neural network technique, genetic algorithm, and sequential quadratic programming for optimization. Zulqurnain Sabir *et al.* [19] proposed a study using Artificial Neural Network (ANN) to solve functional differential models, integrating global optimization techniques PSO and local search optimization techniques SQP. Muhammad Umar *et al.* [20] developed a computational framework to analyze the behavior of PPM using ANN, GA, and the IPA, focusing on optimization and DE development.

The LV competitive model, a basic depiction of species increase and decrease, has been solved using various numerical techniques in existing literature [21]. The Differential Transformation Method (DTM) is a highly effective technique for solving nonlinear equations, surpassing alternative techniques like variational iteration and Adomian decomposition approach [22]. In [23] the study developed perturbation-iteration techniques to accurately approximate Lotka-Volterra system solutions for first-order differential equation systems without requiring a small parameter assumption. FA Abdullah *et al.* [24] utilized numerical methods like Euler Method, Taylor Series, and RK method to analyze species to species competition impact, with the RK Method providing the most accurate solution. [25] examined that Runge-Kutta method outperformed the Laplace Adomian Decomposition Method in finding solutions of DE models related to population dynamics. In [26] wavelet-based approaches are reviewed for solving linear and nonlinear fractional DE, highlighting their efficiency, accuracy. In [27] the proposed MWNN-GAIPA model, which integrates Morlet wavelet NN with genetic and interior-point algorithms, is tested for accuracy, stability, and convergence through multiple test problems and neuron variations. [28] this study enhanced the prediction capability of wavelet neural networks (WNNs) by combining various wavelet families, demonstrating significant improvements in accuracy and efficiency compared to current techniques.

CHAPTER 2

BASIC CONCEPTS AND DEFINITIONS

Chapter 2 provides a comprehensive overview of the key topics relevant to our research area. These foundational topics are crucial for understanding the fundamental concepts and phenomena underlying the study. This chapter also covers areas such as differential equations, their classifications, and an array of methods for solving differential equations, artificial intelligence, neural network, and various types of neural networks. This comprehensive introduction provides the necessary foundation for the research

2.1 Differential Equations

A DE is formed when a mathematical variable and its derivatives are utilised to describe a natural law. Differential equations can describe almost all systems that undergo change. A DE is a mathematical representation that outlines the relationship between an unknown function, its derivatives, independent variables, and related constants [29].

2.2 Classification of Differential Equations

The DE are categorized into various types.

2.2.1 Ordinary Differential Equations (ODE)

An ODE relates an independent variable to a dependent variable along with one or more derivatives of the dependent variable with respect to the independent variable only [30].

The most general expression of an ODE of nth order is given by

$$y^{(n)} = f\left(x, y, y', y'', \ldots, y^{(n-1)}\right), \tag{2.1}$$

which is referred to as ordinary because it involves only one independent variable.

2.2.2 Partial Differential Equations (PDE)

PDE is a system of equations characterized primarily by the presence of multiple independent variables and a single dependent variable [31].
The mathematical representation of nth order partial differential equation is written as:

$$F\left(x_1, x_2, \ldots, x_n, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \ldots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1^2}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \ldots\right) = 0. \tag{2.2}$$

2.2.3 Differential Algebraic Equation (DAE)

DAE represents an extended version of ODE used in mathematical modelling of scientific and engineering problems like in-compressible fluids, optimal control, and chemical process control. The representation of a DAE is given as:

$$F(t, x(t), \dot{x}(t)) = 0. \tag{2.3}$$

2.2.4 Delay Differential Equations (DDE)

A DDE is a type of functional differential equation that involves determining the derivative of an unknown function based on previous values, requiring knowledge of both current and previous states. The general form is given as :

$$\frac{d}{dt}x(t) = f(t, x(t), x(t-\tau)). \tag{2.4}$$

2.2.5 Stochastic Differential Equations

SDE are a powerful framework for simulating dynamic systems influenced by noise or uncertainty, unlike Ordinary Differential Equations (ODEs), which describe deterministic systems. SDEs are important in various disciplines like physics, biology, and finance, describing particle motion, studying population dynamics, and modeling stock prices and interest rates. The general form is given as :

$$\frac{d}{dt}x(t) = f(t, x(t)) + \sigma(t, x(t))\, dW(t). \tag{2.5}$$

2.3 Types of Differential Equation Problems

There are two categories of differential equation problems. These are listed as follows:

2.3.1 Initial Value Problem

An IVP arises when an equation includes a dependent variable, its potential derivatives, and an independent variable, typically time. The problem is defined by assigning specific values of the dependent variable at a particular point in the independent variable's domain. Initial value problems involve time-dependent equations where the dependent variable and its derivatives are specified at the same initial point of the independent variable [32].

2.3.2 Boundary Value Problem

A BVP is a situation in which the dependent variable and its possible derivatives are determined at the extreme of the independent variable. Boundary value problems (BVPs) specify equations or values for solution components at multiple x, with infinite or finite solutions. Programs for BVPs require users to guess the intended solution, often requiring certain parameters and addressing infinite intervals and singularities in coefficients [33].

Dirichlet Boundary Condition

The Dirichlet problem is based on a Dirichlet boundary condition, which outlines the values a solution must take along the domain boundary.

Example:

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega \\ u(x) = g(x), & x \in \partial\Omega \end{cases} \tag{2.6}$$

Neumann Boundary Condition

Neumann boundary conditions are important for solving partial differential equations, using the solution's derivative in contrast to Dirichlet conditions, which define the solution's value at the boundary. These conditions define the derivative of a solution at the boundary of the domain. The Neumann boundary value problem is given as

$$\begin{cases} -\Delta u(x) = f(x), & x \in \Omega \\ \frac{\partial u}{\partial n}(x) = h(x), & x \in \partial\Omega \end{cases} \tag{2.7}$$

where $\frac{\partial u}{\partial n}(x)$ represents the normal derivative of u on the boundary $\partial\Omega$, $h(x)$ is a given function defining the flux or rate of change across the boundary $\partial\Omega$.

Mixed Boundary Condition

Mixed boundary conditions, also known as Cauchy boundary conditions, combine Dirichlet and Neumann boundary conditions, requiring the normal derivative and differential equations values to both take on the domain boundary [32].

The mixed boundary conditions are given by:

$$\begin{cases} \frac{\partial^2 \phi}{\partial \theta^2} + \frac{\partial^2 \phi}{\partial t^2} = f(x,y), \\ \omega_1 \frac{\partial \phi}{\partial n}(0,t) + \omega_2 \phi(0,t) = \omega, \\ \omega_1 \frac{\partial \phi}{\partial n}(L,t) + \omega_2 \phi(L,t) = \omega_i, \end{cases} \tag{2.8}$$

where ω_1, ω_2 are constants, $\frac{\partial \phi}{\partial n}$ denotes the normal derivative at the respective boundaries, ω, ω_i are known constants, L represents the spatial extent of the domain.

2.4 Numerical Methods

Numerical approximation involves using various methods to estimate solutions to differential equations. In mathematics, numerical approximation serves as a vital tool in solving differential equations when direct analytical solutions to differential equations prove difficult or unachievable. Various approaches have been designed to solve differential equations, for example the Euler method [34], RK method [35], the Taylor method [36], and the Finite Difference method [37] enable the computation of solutions for complex physical phenomena by dividing the problems into smaller, more manageable elements or particles. These techniques convert the equations into a solvable format by dividing them and using difference equations to approximate the derivatives. The use of numerical methods is a crucial role in solving DE, providing results through iterative calculations. They involve approximating solutions that may not be exact but are close to the true solution. Numerical methods, while essential in various fields like structural engineering [38], accelerator physics [39], and reaction-diffusion systems [40], have limitations that must be acknowledged.

2.4.1 Euler Method

Euler's method is the most basic and oldest technique to solve IVP, offering a straightforward approach to approximate solutions without requiring advanced algebraic computations. The purpose of Euler's method is to provide approximate solutions to well posed initial value problems [41].

Taylor method is used to derive the Euler method. The Euler method is given as

$$y_{i+1} = y_i + hf(t_i, y_i), \quad \text{for each } i = 0, 1, \ldots, N-1 \tag{2.9}$$

Consider an IVP

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha, \tag{2.10}$$

approximations of y will be calculated at a certain points, known as mesh points within the interval $[a, b]$ which can then be used to estimate the solution at other points with in the interval as continuous approximation of the solution $y(t)$ will not be achieved.

The mesh points are evenly distributed throughout the interval $[a, b]$, achieved by selecting a positive integer N and defining them appropriately.

$$t_i = a + ih, \quad \text{for each } i = 0, 1, 2, \ldots, N.$$

The step size is a typical common separation between positions, calculated as

$$h = \frac{b-a}{N} = t_{i+1} - t_i.$$

2.4.2 Taylor Series

For many years, the Taylor series expansion approach has been employed to find a solution for IVP in ordinary differential equations. The Taylor series technique is a method that uses an infinite series of a function's derivatives to solve initial value problems.

A function's Taylor series is given by

$$y_{n+1} = y_n + y_n^{(1)} \frac{h}{1!} + y_n^{(2)} \frac{h^2}{2!} + y_n^{(3)} \frac{h^3}{3!} + y_n^{(4)} \frac{h^4}{4!} + \cdots, \tag{2.11}$$

where $y_n^{(1)}$, $y_n^{(2)}$, $y_n^{(3)}$, $y_n^{(4)}$ are the first, second, third and fourth derivatives of the function. The differential equation can be solved iteratively using this expansion up to a final value of the independent variable.

The largest derivative of each equation should be retained on the right side in the same order as the required numerical algorithm. To achieve fourth-order numerical algorithms, it is necessary to incorporate derivatives up to and including the fourth-order in the expansions.

The Taylor series method is efficient, accurate, and expandable by maintaining higher-order terms in Taylor expansions, requiring less calculation time [42].

2.4.3 Runge-Kutta Method

RK Method is widely used to solve DE. Taylor procedures, despite their high-order accuracy, require derivative computation and evaluation. Taylor methods are often avoided in practical applications due to their complexity and computational cost. RK techniques provide high-order accuracy similar to Taylor techniques, but they avoid the need to calculate derivatives of $f(t,y)$. The RK technique of the fourth order is a widely known constant-step procedure.

General form of RK-4 is given by

$$\begin{aligned} k_1 &= hf(t_n, y_n), \\ k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 &= hf(t_n + h, y_n + k_3), \\ y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \tag{2.12}$$

The main task in implementing the RK methods is the evaluation of f.

2.4.4 Shooting Method

This method changes boundary value problems into two initial value problems by iteratively adjusts conditions until desired requirements are met at the other end. The given solution to the boundary value problem is obtained by combining the solutions of two corresponding initial value problems. These solutions of the initial value problem are derived using methods such as the Runge-Kutta and Taylor series techniques.

2.4.5 Finite Difference Method

The FDM solves differential equations by approximating continuous derivatives in both the equations and boundary conditions with finite differences. It then solves the resulting system of linear equations using standard techniques to obtain an appropriate finite difference approximation of the derivatives.

Example

Consider BVP,

$$y'' = p(x)y' + q(x)y + r(x), \quad \text{for } a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta. \tag{2.13}$$

To apply the finite difference approach to the linear second-order BVP, both y' and y'' must be approximated using difference-quotient approximations [41].

Forward Difference Approximation

$$y'(x) \approx \frac{y(x+h) - y(x)}{h}. \tag{2.14}$$

Backward Difference Approximation

$$y'(x) \approx \frac{y(x) - y(x-h)}{h}. \tag{2.15}$$

Central Difference Approximation

$$y'(x) \approx \frac{y(x+h) - y(x-h)}{2h}. \tag{2.16}$$

2.5 Artificial Intelligence

AI focuses on the advancement and execution of computer systems designed to solve problems that usually need human expertise. AI creates human intelligence in computer systems, enabling tasks like machine vision, natural language processing, and decision-making systems, despite the complexity of the process [43]. AI is increasingly integrating into various sectors, including government, business, and banks, combining knowledge-based reasoning with complementary methods from other AI domains. [44]. The advancement of AI also has influence on various fields of mathematics.

2.6 Artificial Neural Networks (ANNs)

An ANN is a computational framework that mimics the structure and function of biological neural networks found in the brain. Artificial neurons are interconnected nodes that make up an ANN. These artificial neurons are the fundamental building blocks that make up an artificial neural network. Artificial neuron mimic the action of a biological neuron, i.e., by accepting many different signals x_i, from many nearby neurons and to process them in a simple way. The neuron j determines either to produce an output signal y_j or not, depending on the outcome of processing [45]. Artificial intelligence methods are effective in mathematical problem settings, particularly in inverse problems in imaging sciences and numerical analysis of partial differential equations in high-dimensional regimes [46].

2.6.1 History of Neural Network

The evolution of neural networks occurred in a pattern of step-wise dramatic improvements. In 1943, McCulloch and Pitts developed an algorithm that learns by mimicking the functionality of the human brain and created an artificial neurons that connect and arrange in multiple layers to form artificial neural networks [47]. Rosenblatt created the first perceptron learning algorithm in the late 1950s, followed by Widrow and Hoff's electronic circuit learning rule, igniting active research on artificial neural networks in the 1960s. Minsky and Papert's 1969 book Perceptrons highlighted the computational limitations of single-layer neural networks, leading to a decline in funding for artificial neural network research. The "golden decade" of NNs began in the 1960s and 1950s, followed by a period of peace in the 1970s. In 1986, the PDP research group published Parallel Distributed Processing texts, which introduced MLPs and the backpropagation learning method, enabling the training of multiple layers of perceptrons in artificial neural networks [48]. The INNS was established in 1987, leading to the IEEE International Conference on Neural Networks, IEEE Transactions on NN, Neural Computation, and the magazine Neural Networks [32].

2.7 Mathematical Model of Artificial Neural Network

A neuron N_i receives a set of n inputs , $S = \{x_j \mid j = 1, 2, \ldots, n\}$. Before reaching the main body of a neuron, each input to a neuron N_i is multiplied by a weight factor w_{ij} for $j = 1, 2, \ldots, n$. The neuron requires a bias term b_0 and a threshold value θ_k to generate an output signal. The generated weight signal is influenced by a function called the activation function.
In terms of mathematics, the i^{th} neuron N_i output is given by

$$O_i = f\left[b_0 + \sum_{j=1}^{n} w_{ij}x_j\right]. \tag{2.17}$$

2.8 Activation Functions

ANN use activation functions to convert input signals into output signals, which are then sent to subsequent layers in the stack, after determining the sum of input products and weights. Mathematically we can write it as;

$$\text{net} = w_{i1}x_1 + w_{i2}x_2 + \cdots + w_{ij}x_j + \theta, \tag{2.18}$$

where θ is a threshold value that is added to the neurons. The threshold-based classifier is crucial in activation functions, determining whether a neuron is deactivated or activated based on the input value exceeding a threshold, preventing output from being sent to the next layer. The accuracy of the prediction of the neural network depends on the number of layers used and the type of activation function used. Nonlinear activation functions are common in neural networks. NN behaves like a linear regression model where the predicted output is the same as the input provided if an activation function is not defined [49].

2.8.1 Sigmoid Activation Function

The sigmoid activation function converts the input range from $(-\infty;+\infty)$ to the range in $[0;1]$. It is a non-linear and smooth in nature. The sigmoid's [0;1] output range compresses unit output, making gradient disappear in deep networks, making network improvement difficult over time [50].
Mathematically Sigmoid function is given as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{2.19}$$

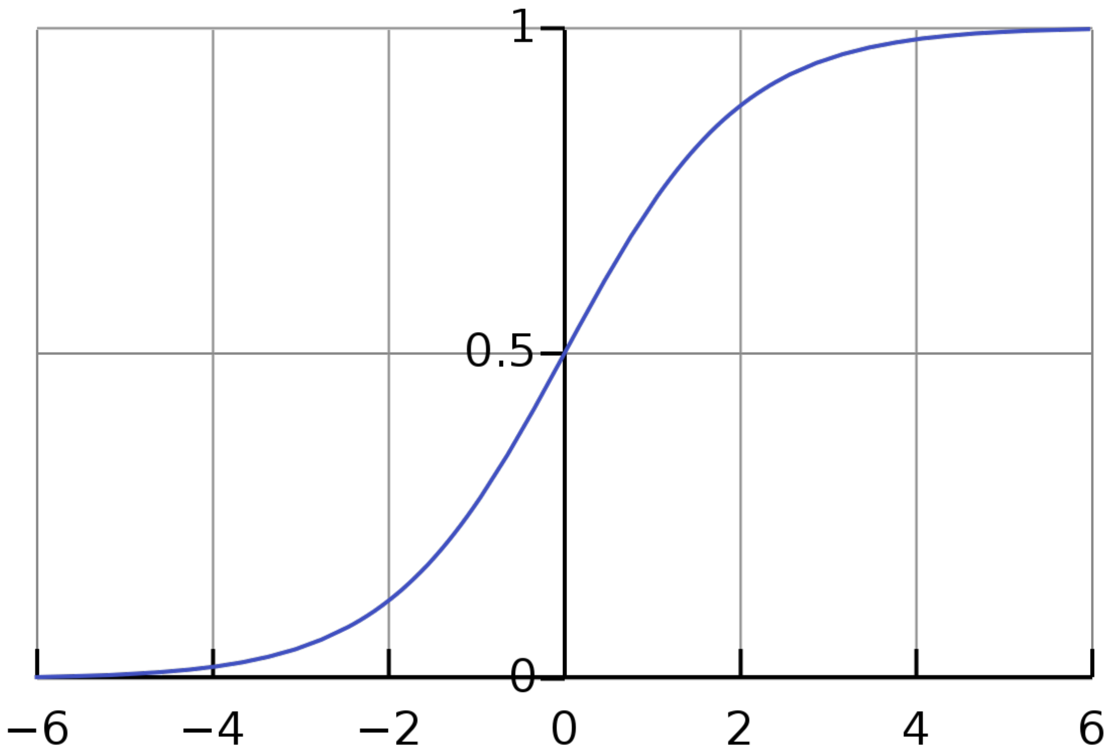

Figure 2.1: Sigmoid Activation Functions

2.8.2 Binary Step Function

Binary Step Function is the most simple activation function and it can be implemented with simple if-else statements in Python. Binary activation functions are commonly used in binary classifiers, but they cannot be applied when the target carriage has multiclass classification. The binary step function's gradient is zero, potentially affecting the backpropagation step, as its derivative equals zero when computed with respect to x.
Binary step function is written as

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2.20}$$

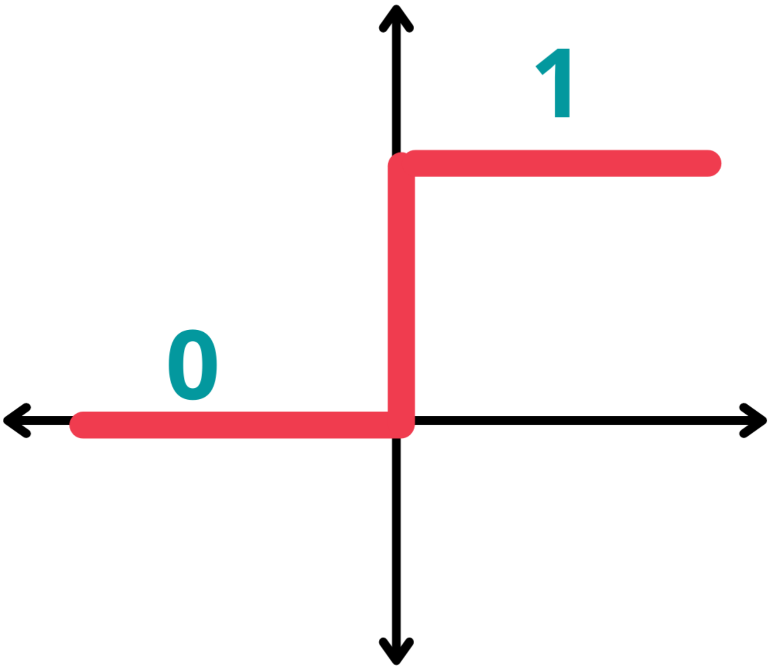

Figure 2.2: Binary Step Activation Functions

2.8.3 Linear Activation Function

The linear activation function is directly proportional to the input. It overcomes the zero gradient issue in binary step functions by defining it as:

$$F(x) = ax. \tag{2.21}$$

Value of variable a can be a constant value. In this case, the value of derivative of the function $f(x)$ is equal to constant that is utilized. The gradient is constant value not a zero, independent of the input value x ,indicating that weights and biases will change during the backpropagation step [49].

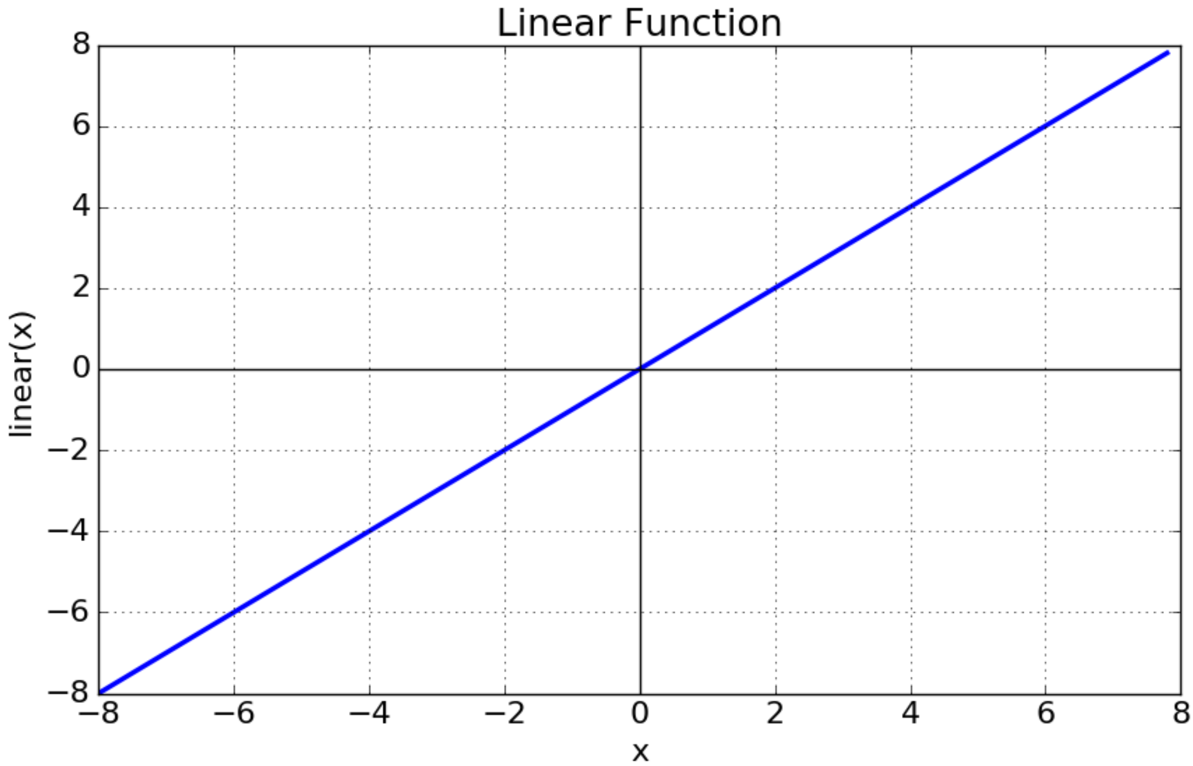

Figure 2.3: Linear Activation Functions

2.8.4 Morlet Wavelet Activation Function

Wavelets are reffered to as functions with zero integral value, localized along the time axis, that can shift and scale. The MW function, a fundamental function, is utilized in the model to effectively address optimization issues [51].
MW function is described as follows:

$$f(x) = \cos(1.75t)e^{-0.5t^2}. \tag{2.22}$$

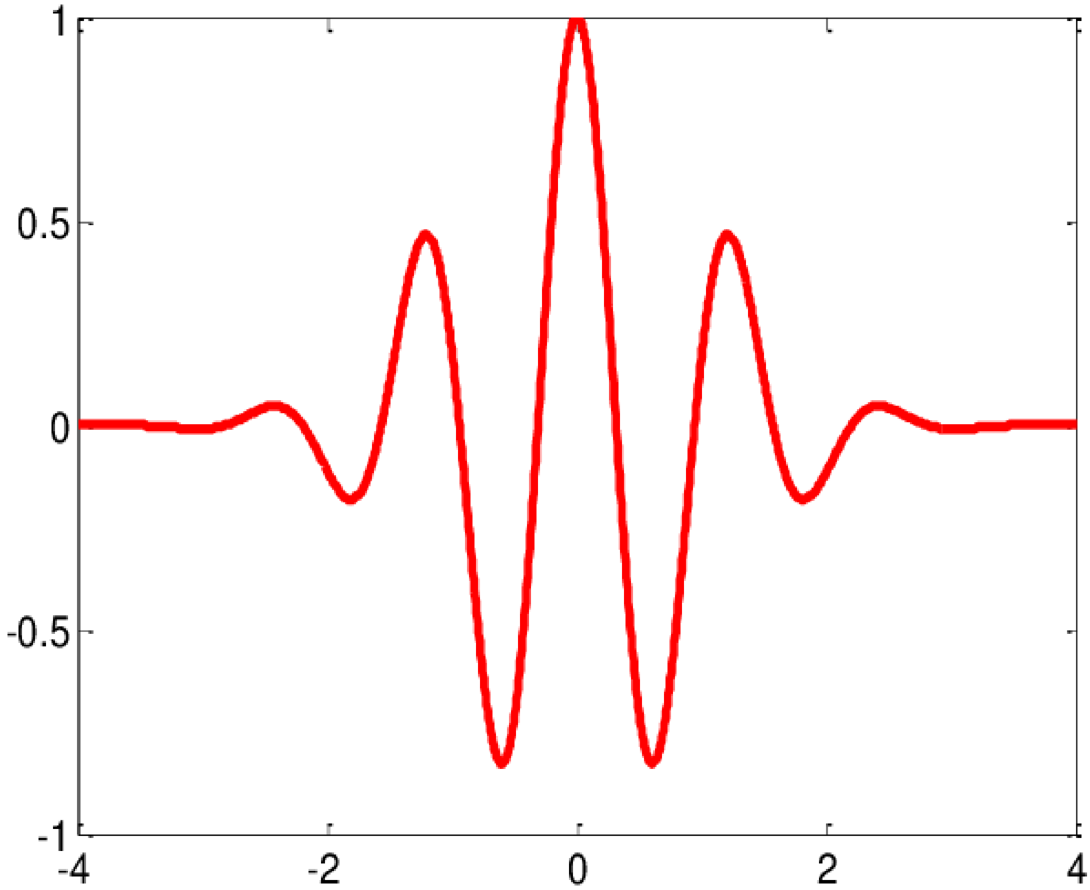

Figure 2.4: Morlet Wavwlet Activation Functions

2.8.5 Mexican Hat

Mexican Hat activation function is a helpful optimization tool for solving differential equations, particularly when utilizing Artificial Neural Networks (ANNs). The shape resembles a "hat" or bell curve, effectively captures localized features in data, making it ideal for modeling oscillatory and wave-like phenomena. The Mexican Hat function enhances neural networks' ability to approximate solutions, including intricate patterns like wave propagation or damped oscillations, in differential equations.

Mexican hat Function mathematically can be written as

$$f(x) = \frac{2}{\sqrt{3}}\pi^{-\frac{1}{4}}(1-t^2)e^{-\frac{t^2}{2}}. \tag{2.23}$$

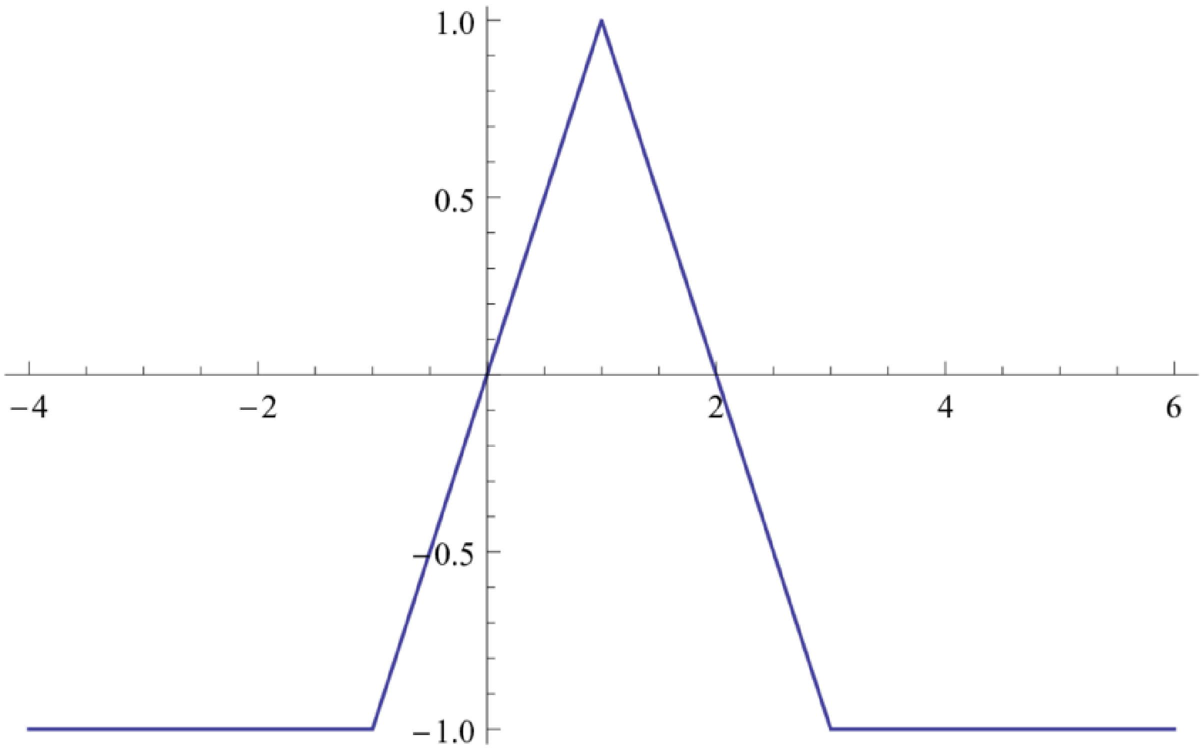

Figure 2.5: Mexican Hat Activation Functions

2.9 Neural Network Artitecture

In practical scenarios, a single node is often inadequate, making networks with multiple nodes a more common choice. The connections between nodes are crucial in the early design of neural networks, as they determine the execution of calculations. An ANN is a data processing system composed of numerous simple, densely connected neurons. An artificial neuron functions similarly to the biological neurons they accept and processes signals from nearby neurons and decides whether to fire an output signal based on the processed results [52].

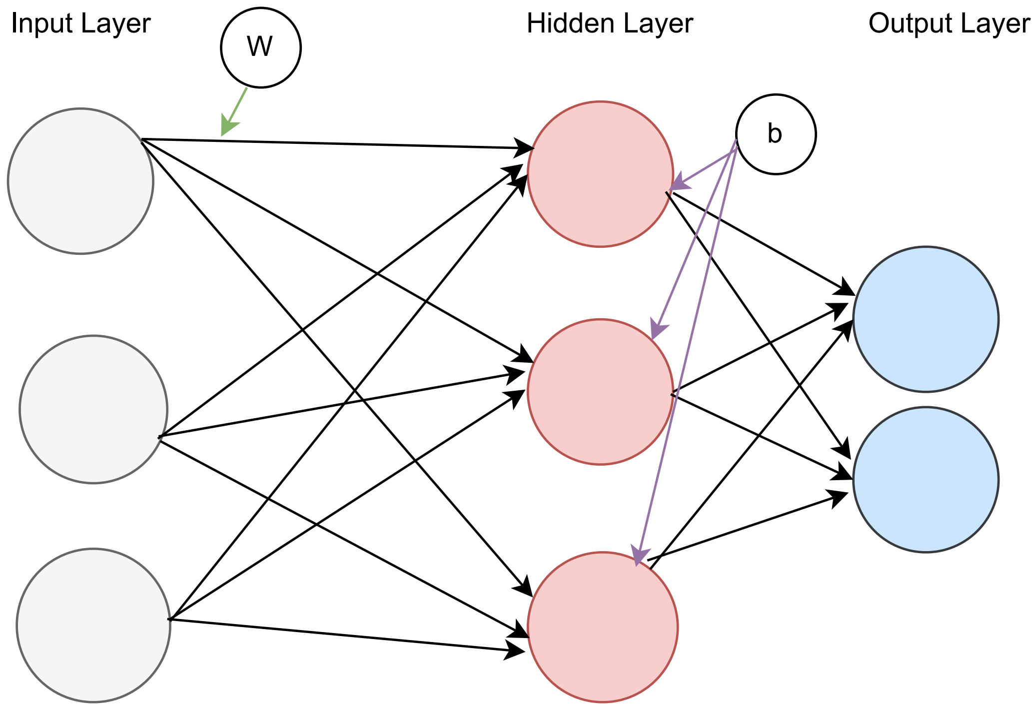

Figure 2.6: Architecture of ANN

2.9.1 Feed Forward Neural Networks

A NN is classified as a feed forward neural network when there is no feedback loop, meaning the outputs of the neurons do not feed back into the inputs within the network. This network only allows forward information flow from input nodes to output nodes through hidden nodes within the network. NN are often arranged in layers, and feed forward neural networks are further divided into single-layer and multilayer networks according to the number of layers [53]. Graphical representation is given in Fig. 2.7.

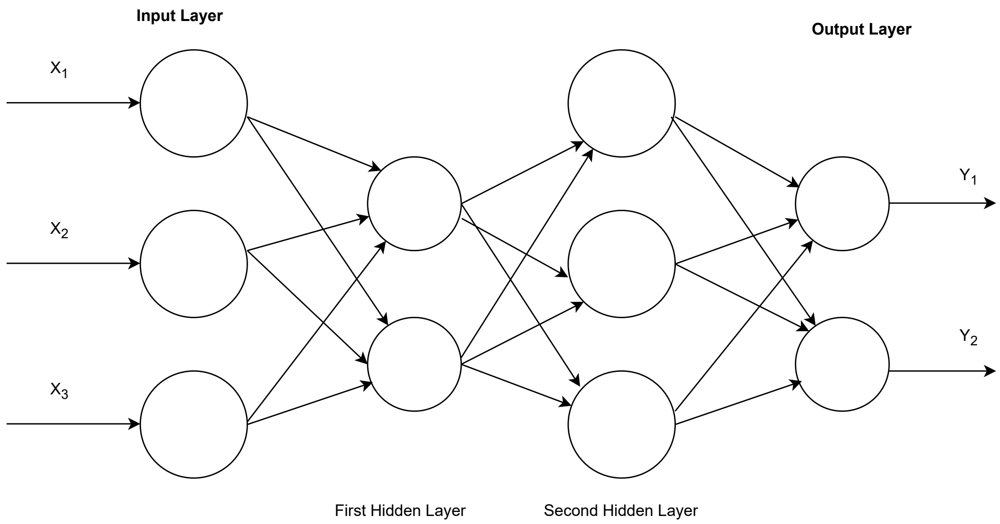

Figure 2.7: Feed Forward Neural Network

2.9.2 Recurrent Neural Networks

A network is referred as a recurrent neural network if feedback is present, such as synaptic connections from outputs to inputs (either to the same neurons or to others). This approach is highly beneficial when solving problems that rely on both current and previous inputs. The recurrent network transfers data from inputs to outputs while learning, and vice versa, until the output numbers remain constant [32].

2.9.3 Radial Basis Function Neural Network (RBF-NN)

In a RBF network, the structure starts with an input layer, followed by a hidden layer containing the basis functions, and ends with an output layer.

2.10 Learning in Neural Networks

Machine learning focused on designing and building algorithms and techniques that helps computers to "learn" from data and improve their performance over time. Two widely recognized types of learning are supervised learning and unsupervised learning [54].

2.10.1 Supervised Learning

In supervised learning the system is provided with the desired output. A supervised learning approach involves altering weights by comparing them to a set of goal outputs. The algorithm generates a mathematical model in the learning process by utilizing a comprehensive data set that containing all inputs and outputs. Text instances are utilized to teach algorithms, with pre-defined input and output. The algorithm in this study receives the input set and the corresponding results. The algorithm compares its actual results with the relevant outcomes to obtain the results [55]. The inaccuracy is determined by comparing the network's calculated output with the predicted output upon correction. After the error identification performance can be enhanced by doing changing in network settings.

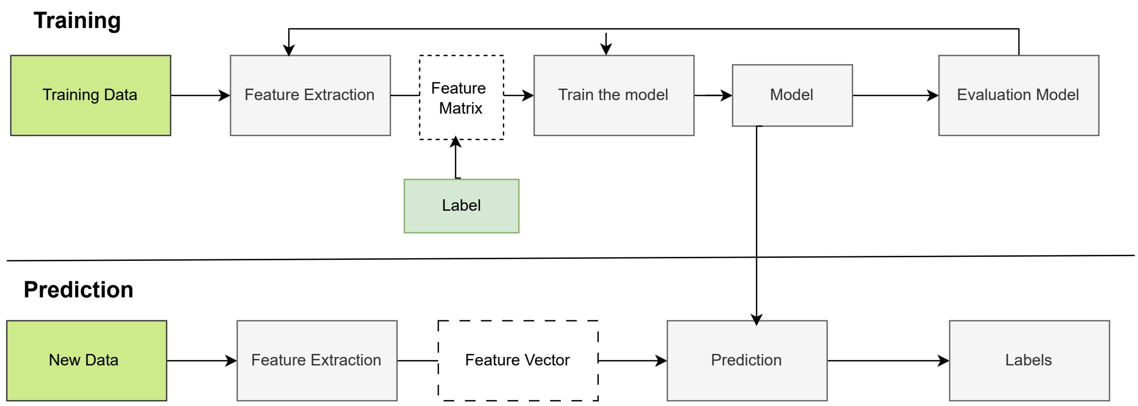

Figure 2.8: Supervised Learning workflow

2.10.2 Unsupervised Learning

In unsupervised learning, the system is presented only with the input data, and the aim is to uncover the inherent patterns and structure within the data. In an unsupervised learning approach, weight adjustments are not based on a comparison with a target output. In this approach, there is no direct teaching signal for weight adjustments; however, it still requires certain guidelines to facilitate successful learning. This property is referred to as self-organization.

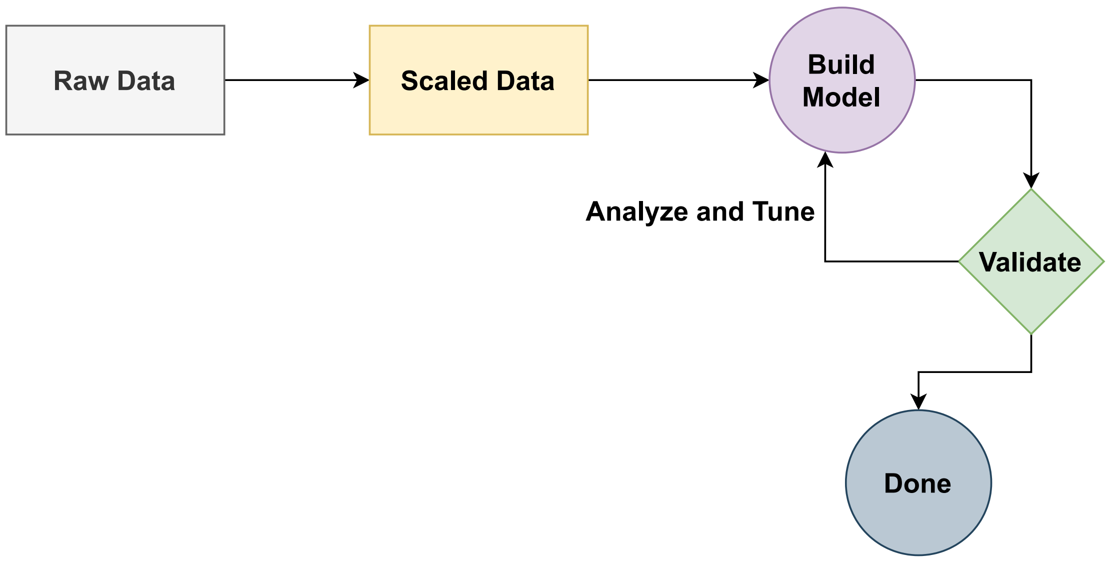

Figure 2.9: Unsupervised Learning workflow

2.11 ANN in solving Differential Equation

Artificial neural network appeared as a effective tool for solving differential equations, mainly those that are complex or lack analytical solutions. ANN in association with optimization tools offers an alternative approach to traditional numerical methods for solving ODE, PDE, and nonlinear DE. ANN techniques have effectively given the solutions of linear/nonlinear systems such as an algorithm based on neural network to solve differential equation of fractional order [56], utilizing artificial neural networks to solve fractal-fractional differential equations [57], an adaptive algorithm to solve fractional partial differential equations using wavelet artificial neural networks [58], fuzzy differential models [59], singular Lane–Emden model [60], nonlinear Jeffery–Hamel flow model [61].

CHAPTER 3

REVIEW OF COMPARATIVE ANALYSIS OF TAYLOR SERIES AND RUNGE-KUTTA FEHLBERG METHODS IN SOLVING THE LOTKA-VOLTERRA COMPETITIVE MODEL

3.1 Introduction

This study compares the Runge-Kutta Method and Taylor Series for sloving mathematical models, focusing on the Lotka-Volterra competitive model. This Lotka-Volterra competition model explains the growth dynamics of Paramecium Caudatum and Stylonychia Pustulata, considering interspecific competitive effects, carrying capacities, and population growth rates. This research employs Mathematica 13.2 for numerical solutions of Two- Species Competition Model to demonstrate how these approaches effectively manage non-linear interactions in the model. This paper highlights the importance of numerical techniques like Runge-Kutta Fehlberg methods and the Taylor Series for approximating solutions in situations where analytical solutions may be impossible or complex.

3.2 Problem statement

The Lotka-Volterra competitive model is a crucial mathematical framework for analyzing biological interactions, particularly competitive dynamics between species, understanding of ecological balance and resource allocation. Traditional methods for solving complex biological equations often face challenges that may hinder their ability to provide precise answers. This review focuses on the effectiveness of numerical techniques like the RK method and the Taylor Series method in overcoming challenges.

3.3 Background of research on Two-species Competition Model

In an ecological system, species interact through factors like climate, resources, population density, and reproduction capacity. These interactions can be categorized as predators, competition, mutualism or parasitism. Biological interaction can have positive, negative or no impact on interacting species. Competition is a type of biological interaction that can be harmful to both species involved. In competition species compete for the shared resource in a specific region. The author's previous publications extensively examined the growth of mixed populations of various species under mutual interdependence, covering numerous special scenarios[62]. Volterra has addressed a unique case of two species competing for a common food supply, which extends and easily provides a solution [63]. Volterra's theory suggests that population growth rate is directly proportional to the current population size without limiting factors, following established literature principles

$$\frac{dN}{dt} = rN. \tag{3.1}$$

leading to a population growth law that is exponential (Malthusian). The natural limitation of food supply transforms the coefficient r into a decreasing function of N. If $r > 0$, the model predicts exponential growth; if $r < 0$, it predicts exponential decrease. In exponential growth a population can increase continuously at an exponential rate. Logistic growth model overcome this prediction of an exponential growth model which is a simplest extension of 3.1

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right). \tag{3.2}$$

It is basically the most basic rule which states that population density determines intrinsic growth rate, explaining the reason behind the decrease in per capita growth rate as population grows [64].

3.4 Two Species Competition Model

The independent contributions of Lotka and Volterra, who developed many model including predator-prey dynamics and two-species competition, significantly influenced the field now called as population biology. They were among the first to explore the interactions between species by introducing several simplifying assumptions, resulting in non-trivial but manageable mathematical problems [65]. The two species competition model is an ecological framework that illustrates species interaction and competition for resources. It helps predict the growth and fall of competing species in population ecology, using a system of nonlinear equations. Life equilibrium is influenced by competition, and understanding species interaction is crucial for predicting outcomes in natural settings.The two-species competition model in population ecology describes the interdependent growth and fall of two competing species using a system of nonlinear equations. The two species competition model is proposed by Lotka-Volterra [64, 66].

Two species competition model is given by:

$$\frac{dx}{dt} = r_1 x\left(1 - \frac{x+\alpha y}{K_1}\right), \tag{3.3}$$

$$\frac{dy}{dt} = r_2 y\left(1 - \frac{y+\beta x}{K_2}\right), \tag{3.4}$$

$$x(0) = x_0, \tag{3.5}$$

$$y(0) = y_0, \tag{3.6}$$

where r_1, r_2 are intrinsic growth rates of species 1 and species 2 respectively. K_1, K_2 are carrying capacities of the environment for species 1 and species 2 respectively. α is Competition coefficient that evaluates the effect of species 2 on species 1. β is Competition coefficient that evaluate the effect of species 1 on species 2. $x = 1$ is initial population of species 1 at time $t = 0$. $y = 1$ is initial population of species 2 at time $t = 0$.

In this research, two numerical methods, the Taylor Series Method and the Runge-Kutta (RK) method, are used for solving the model and their accuracy is compared to Morlet Wavelet based ANN using hybrid ANN optimization techniques Genetic Algorithm and Sequential Quadratic Programming (GA-SQP) and discern which one gives the best results. This thesis aims to investigate the effectiveness of optimization techniques, such as GA-SQP, in achieving better results compared to traditional numerical techniques in particular RK-4 and Taylor Series techniques. We have performed some variations in parameters $(\alpha, \beta, K_1, K_2)$, that are used in equation 4.19 and equation 4.20.

3.5 Methodology

This study primarily focuses on the Lotka-Volterra competitive model, which explains interactions between two species competing for limited resources. The study uses ordinary differential equations to represent a model and compares Runge-Kutta Fehlberg (RKF) and Taylor Series numerical techniques for resolving these equations. The coupled equation, requires a significant amount of time to solve using an exact analytical solution, is numerically solved using Mathematica 13.2.

3.5.1 Taylor Series Method

The Taylor Series Method is a method that transforms a function into an infinite series for approximating the solution of differential equations. This study use Mathematica 13.2 to solve model equations by Taylor series. To utilize the Taylor Series method, it is necessary to first determine the derivatives of model equations. The model equations provide the first derivatives given in 4.19, 4.20. Differentiating these equations give second derivative

$$\frac{d^2x}{dt^2} = r_1 \left(1 - \frac{2x + \alpha y}{k_1}\right) \frac{dx}{dt} - \frac{r_1 \alpha x}{k_1} \frac{dy}{dt}, \tag{3.7}$$

$$\frac{d^2y}{dt^2} = r_2 \left(1 - \frac{2y + \beta x}{k_2}\right) \frac{dy}{dt} - \frac{r_2 \beta y}{k_2} \frac{dx}{dt}. \tag{3.8}$$

This study uses Mathematica 13.2 to solve model equations by Taylor series.

3.5.2 Runge-Kutta Method

The Runge-Kutta technique is a well-known procedure that can accurately approximate a Taylor Series approximation without requiring higher derivative computations. This study utilizes Mathematica 13.2 to solve model equation by RK Method.

$$k_1 = hf(x_i, y_i),$$

$$k_2 = hf\left(x_i + \tfrac{1}{4}h,\ y_i + \tfrac{1}{4}k_1\right),$$

$$k_3 = hf\left(x_i + \tfrac{3}{8}h,\ y_i + \tfrac{3}{32}k_1 + \tfrac{9}{32}k_2\right),$$

$$k_4 = hf\left(x_i + \tfrac{12}{13}h,\ y_i + \tfrac{1932}{2197}k_1 - \tfrac{7200}{2197}k_2 + \tfrac{7296}{2197}k_3\right),$$

$$k_5 = hf\left(x_i + h,\ y_i + \tfrac{439}{216}k_1 - 8k_2 + \tfrac{3680}{513}k_3 - \tfrac{845}{4104}k_4\right),$$

$$k_6 = hf\left(x_i + h,\ y_i - \tfrac{8}{27}k_1 + 2k_2 - \tfrac{3544}{2565}k_3 + \tfrac{1859}{4104}k_4 - \tfrac{11}{40}k_5\right).$$

$$y_{i+1} = y_i + \tfrac{16}{135}k_1 + \tfrac{6656}{12825}k_3 + \tfrac{28561}{56430}k_4 - \tfrac{9}{50}k_5 + \tfrac{2}{55}k_6 \qquad (3.9)$$

Mathematica 13.2 utilizes the built-in `NDSolve` function to construct the RK method.

3.5.3 Comparison of Methods

The accuracy of RK and the Taylor Series method can be assessed by comparing the results from both approaches.

Table 3.1: Numerical solution for P. Caudatum and S. Pustulata.

Days	P. Caudatum in mixed	S. Pustulata in mixed
0	2.000000000	2.000000000
1	3.751451974	2.000000000
2	4.504472679	2.000000000
3	6.808330731	9.304627060
4	11.683631120	16.591242660
5	18.612572770	24.421613580
6	27.522162810	30.003024910
7	38.313686050	32.453811910
8	50.911572770	32.557535140
9	65.061172600	31.279097530
10	80.206496750	29.258908940
11	95.554528050	26.882724210
12	110.267244900	24.399495860
13	123.667222000	21.977749000
14	135.357183700	19.725021040
15	145.223933000	17.698338470
16	153.364466400	15.916144810
17	159.991369700	14.371408780
18	165.355793800	13.043100870
19	169.699790500	11.904482790
20	173.233788400	10.928200410
21	176.130265700	10.088946640
22	178.525818200	9.364558700
23	180.526588800	8.736231420
24	182.214318400	8.188295469
25	183.651802800	7.707823312
26	184.887408900	7.284194294

This study provides essential insights into the application of numerical approximation techniques, including the Taylor Series and RKF method, in tackling the Lotka-Volterra competitive model. Table 3.1 shows that the RKF approach outperformed the Taylor Series by a significant margin. RKF is the more reliable option when biological modeling precision is the top priority. In conclusion, the Taylor Series may be uncomplicated, but it is less appropriate for accurate biological modeling of the organisms in issue due to its decreased accuracy, particularly when contrasted with the RKF approach.

3.6 Novelty of Research Thesis

It is evident from reviewing so many academic publications that authors main focus in current technological age remains on computation intelligence (CI)-based solvers. This research employs a hybrid GA-SQP strategy to optimize Morlet wavelet functions and artificial neural networks (ANN) for solving complex differential equations. The ANN-GA-SQP framework is created to solve Two-Species Competition Model and obtain the best solution.

CHAPTER 4

DESIGN OF MORLET WAVELET ARTIFICIAL NEURAL NETWORK FOR SOLVING TWO-SPECIES COMPETITION MODEL

4.1 Introduction

This chapter focuses on solving Two- Species Competition Model using a hybrid Artificial Neural Network (ANN) aiming to enhance the accuracy, stability, and adaptability in differential equations. The hybrid approach utilizes local optimizer SQP and global optimizer GA. The Morlet wavelet is used as an activation function for the modeling of ANN to solve the model. The effectiveness of a strategy is verified by comparing numerical solutions from traditional techniques such as the RK method and the Taylor series method. The model's non-linearity and sensitivity to parameter changes make it challenging to manage and find analytical solutions. Conventional numerical techniques such as the RK4 and Taylor series can approximate solutions, but their precision requires small time steps, increasing computing costs. ANN-based methods offer a viable substitute and their accuracy and stability require careful optimization. The hybrid GA-SQP optimization approach improves the accuracy of the ANN model while maintaining computing efficiency.

The hybrid GA-SQP strategy aims to combine the advantages of local and global optimization methods. The Genetic Algorithm (GA) is utilized as a global optimizer to determine the optimal initial weights for the ANN to ensure a well-optimized model start. GA may not always provide the maximum level of accuracy needed to solve differential equations. To overcome the limitation, local optimizer SQP is used to enhance the convergence of the GA-obtained solution [9]. SQP is a highly effective local optimization technique, especially for constrained nonlinear optimization problems. SQP is effective in achieving rapid convergence to an ideal solution by fine-tuning solutions within a local region, unlike GA which explores the entire search space. The hybrid GA-SQP optimization approach ensures that the ANN accurately and efficiently represents the Two-Species Competition Model. It is more effective in comparison to other numerical techniques, as it improves accuracy and also reduces the computational burden. The Morlet wavelet activation function enhances the network's ability to approximate complex, nonlinear differential equations.

The main purpose of the study is to effectively manage the non-linearity of the model while providing highly accurate solutions for the Two-Species Competition Model. The Morlet wavelet activation function significantly enhances the network's approximation capacity, making it an ideal tool for modeling differential equations. The study highlights the potential of ANN-based techniques as a viable alternative to traditional numerical techniques for solving DE in mathematical and ecological models.The proposed method utilizes hybrid GA-SQP and morlet wavelets to reduce error, enhance stability, and offer a more optimal solution framework.

4.2 Problem Statement

The Two-Species Competition Model is used to model the interaction between two competing species and is often presented by using the coupled nonlinear mathematical model based on differential equations. To understand the dynamics of the system, the accurate solution of these differential equations is crucial. In the Two-Species Competition Model, researchers proposed various numerical solvers for instance Runge-Kutta and Taylor Series [67] but these methods are not generalized, are less accurate, and are computationally expensive. In the recent past, ANN-based numerical solvers have emerged as powerful tools for approximating solutions of various systems based on nonlinear differential equations [68, 69, 70]. However, the designing of the ANN-based numerical solvers integrated the computational efficiency of the Morlet wavelet and optimized with hybrid GA-SQP for the approximating solutions of the Two-Species Competition Model is challenging. The Morlet wavelet-based ANN is a promising alternative to investigate the dynamics of nonlinear coupled systems. This study designs and evaluates the Morlet wavelet-based ANN optimized with hybrid GA-SQP for the numerical treatment of the Two-Species Competition Model, aiming to improve the accuracy and stability.

4.3 Research Gap

Despite advancements in ANN-based differential equation solvers and their potential in solving various differential equation types, Artificial Neural Networks (ANNs) have not yet tackled the Two-Species Competition Model. No research has utilized wavelet-based ANNs, particularly those utilizing the Morlet wavelet activation function, on this ecological model. The integration of hybrid optimization techniques like GA-SQP for such networks remains a subject of further research. This study presents a novel ANN framework optimized with GA-SQP and the Morlet wavelet to solve the Two-Species Competition Model, aiming to fill a crucial gap.

- The effectiveness of hybrid GA-SQP optimization for ANN-based solvers in ecological models has been under-researched.
- Further research is required to explore the potential of wavelet-based activation functions, specifically Morlet wavelet, in improving the performance of Artificial Neural Networks (ANN).

4.4 Research Objective

The research objective to address existing knowledge gaps is as follows.

- To develop a novel ANN framework utilizing the computing efficiency of Morlet wavelet along with hybrid optimizing the performance of GA-SQP for solving differential equations based on Two-Species Competition Model.
- To compare the accuracy and stability of the Morlet wavelet-based ANN with existing numerical techniques for solving the two-species competition model.

4.5 Methodology

Morlet Wavelet-ANNs is designed for solving the Two-Species Competition Model. Morlet Wavelet activation function is used to build the fitness function, and optimization is performed by GA integrated with SQP. The methodology is represented graphically in Fig.4.1.

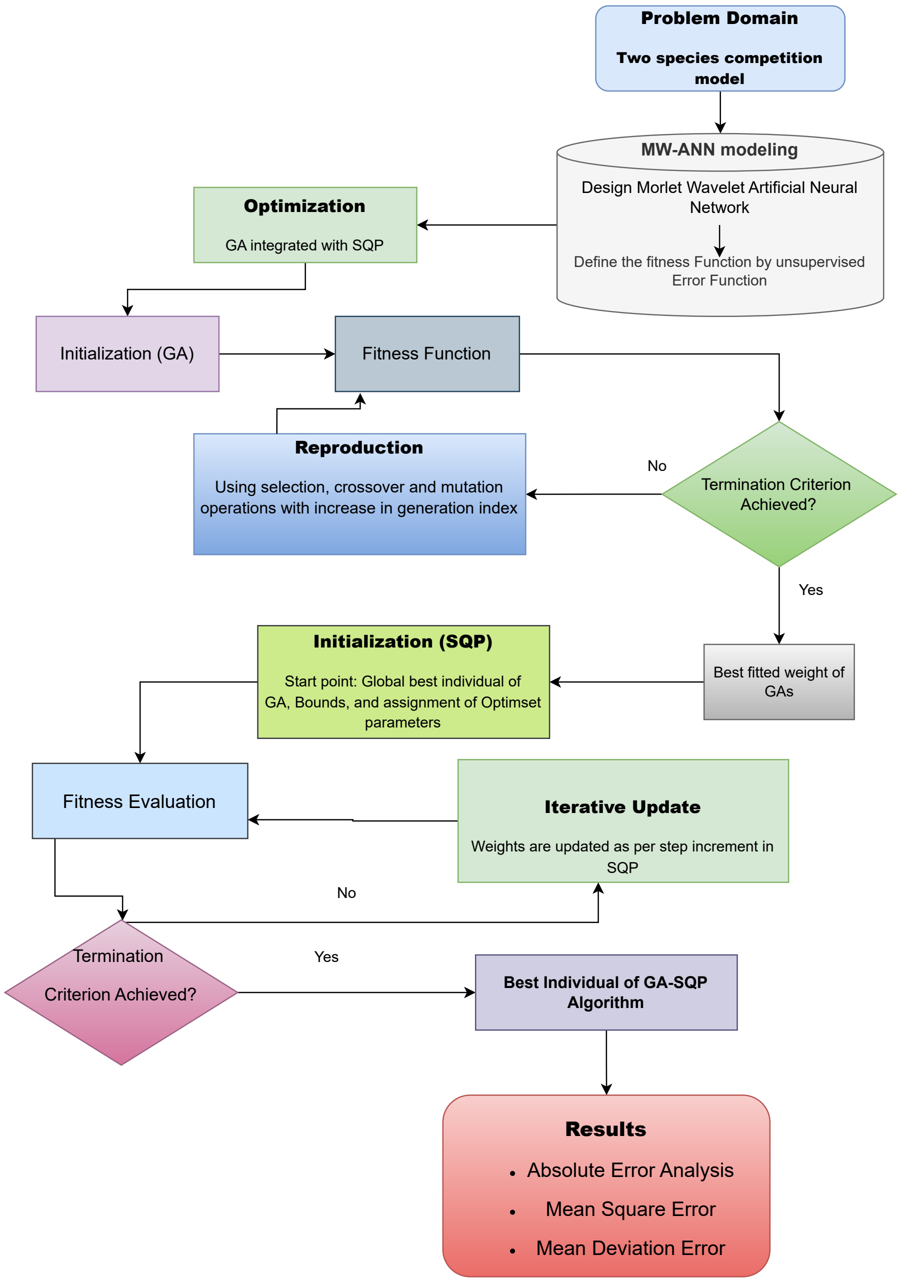

Figure 4.1: Methodology of ANN

4.5.1 Design of MW-ANN

ANNs are known for their ability to provide accurate and reliable answers in various applications [71, 72]. Mathematical formulations of Two-Species Competition Model given in the system 4.19 and 4.20 using feed-forward Morlet Wavelet-ANNs in terms of approximate outcomes and its 1st order derivative is given as:

$$[\hat{x}(t)] = [\sum_{i=1}^{n} \alpha_{ix} f(W_{ix}t + b_{ix})]. \tag{4.1}$$

$$[\hat{x}'(t)] = [\sum_{i=1}^{n} \alpha_{ix} f'(W_{ix}t + b_{ix})]. \tag{4.2}$$

$$[\hat{y}(t)] = [\sum_{i=1}^{n} \alpha_{iy} f(W_{iy}t + b_{iy})]. \tag{4.3}$$

$$[\hat{y}'(t)] = [\sum_{i=1}^{n} \alpha_{iy} f'(W_{iy}t + b_{iy})]. \tag{4.4}$$

where n represents the number of neurons and unidentified weight vectors are presented by [α, w, b], *i.e.*

$W = [w_x, w_y]$, $b = [b_x, b_y]$ and $\alpha = [\alpha_x, \alpha_y]$

$b_x = [b_{1x}, b_{2x}, b_{3x}, \ldots, b_{kx}]$, $b_y = [b_{1y}, b_{2y}, b_{3y}, \ldots, b_{ky}]$

$\alpha_x = [\alpha_{1x}, \alpha_{2x}, \alpha_{3x}, \ldots, \alpha_{kx}]$, $\alpha_y = [\alpha_{1y}, \alpha_{2y}, \alpha_{3y}, \ldots, \alpha_{ky}]$

$w_x = [w_{1x}, w_{2x}, w_{3x}, \ldots, w_{kx}]$, $w_y = [w_{1y}, w_{2y}, w_{3y}, \ldots, w_{ky}]$

The design of MW-ANNs will be presented for solving the Two-Species Competition Model. This function is given as

$$f(x) = \cos(1.75t)e^{-0.5t^2}, \tag{4.5}$$

$$f(W_i t + b_i) = \cos(1.75(W_i t + b_i))e^{-0.5(W_i t + b_i)^2}. \tag{4.6}$$

Using the Morlet Wavelet as a activation function, the updated form of (4.1-4.4) is given by

$$\hat{x}(t) = \sum_{i=0}^{n} \alpha_i \left(\cos(1.75(W_i t + b_i))e^{-0.5(W_i t + b_i)^2} \right). \tag{4.7}$$

$$\hat{y}(t) = \sum_{i=0}^{n} \alpha_i \left(\cos(1.75(W_i t + b_i))e^{-0.5(W_i t + b_i)^2} \right). \tag{4.8}$$

$$\hat{x}'(t) = \sum_{i=1}^{n} -\alpha_i W_i e^{-0.5(W_i t + b_i)^2} \left[1.75 \sin(1.75(W_i t + b_i)) + 2(0.5)\cos(1.75(W_i t + b_i))(W_i t + b_i)\right]. \tag{4.9}$$

$$\hat{y}'(t) = \sum_{i=1}^{n} -\alpha_i W_i e^{-0.5(W_i t+b_i)^2}\left[1.75\sin(1.75(W_i t+b_i)) + 2(0.5)\cos(1.75(W_i t+b_i))(W_i t+b_i)\right]. \tag{4.10}$$

The graphical representation of methodology is given in 4.2

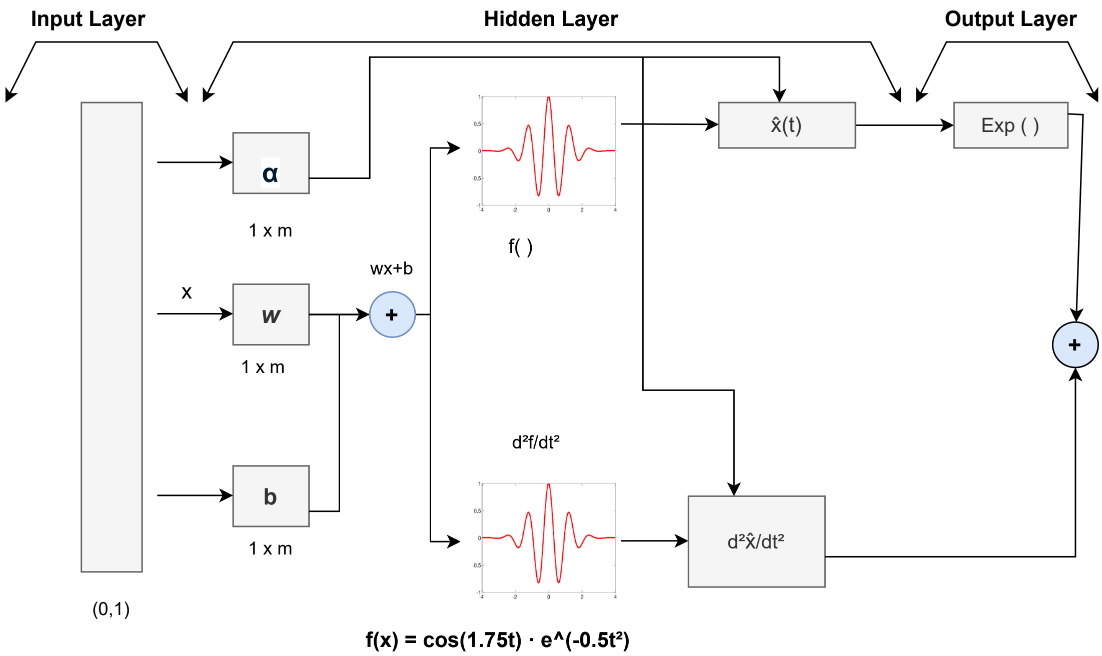

Figure 4.2: MWANNs architecture

4.5.2 Justification for selecting the Morlet Wavelet Activation Function

In this work, the Morlet wavelet function is used as the activation function because it can capture the nonlinear and complex dynamics of the Two-Species Competition Model more effectively than common functions like sigmoid, tanh, or ReLU. Traditional functions have certain drawbacks: sigmoid may show good accuracy but fails to maintain stability over short time intervals; tanh and ReLU often cause deviations in solutions when dealing with nonlinear systems. The Morlet wavelet provides better approximation capabilities for nonlinear systems.

4.5.3 Fitness Function

An essential part of optimization algorithms, especially evolutionary techniques like GA, is the fitness function. The fitness function is able to differentiate between better and worse solutions by assessing each candidate's performance using the fitness function. The selection process is guided by this evaluation, which guarantees that individuals with greater fitness have a greater chance of being selected for reproduction, ultimately resulting in an optimal or nearly optimal solution. For solving the two species competition model, the fitness function is formulated

$$\varepsilon_f = \varepsilon_x + \varepsilon_y + \varepsilon, \tag{4.11}$$

where ε_x from equation 4.11 is given as:

$$\varepsilon_x = \left[\frac{d\hat{x}}{dt} - r_1\hat{x}\left(1 - \frac{\hat{x}+\alpha\hat{y}}{k_1}\right)\right]^2. \tag{4.12}$$

ε_y From equation 4.11 is given as:

$$\varepsilon_y = \left[\frac{d\hat{y}}{dt} - r_2\hat{y}\left(1 - \frac{\hat{y}+\beta\hat{x}}{k_2}\right)\right]^2. \tag{4.13}$$

ε from initial conditions 4.11 is given as:

$$\varepsilon = (\hat{x}(0) - x_0)^2 + (\hat{y}(0) - y_0)^2. \tag{4.14}$$

Eq 4.11 becomes

$$\varepsilon_f = \left[\frac{d\hat{x}}{dt} - r_1\hat{x}\left(1 - \frac{\hat{x}+\alpha\hat{y}}{k_1}\right)\right]^2 + \left[\frac{d\hat{y}}{dt} - r_2\hat{y}\left(1 - \frac{\hat{y}+\beta\hat{x}}{k_2}\right)\right]^2 + (\hat{x}(0) - x_0)^2 + (\hat{y}(0) - y_0)^2. \tag{4.15}$$

4.5.4 GA-SQP Optimization

In order to solve the Two-Species Competition Model, an evolutionary computing approach called MV-GA-SQP, which combines Morlet Wavelet (MV) and Genetic algorithm (GA) supported by Sequential Quadratic Programming (SQP), is used to optimize the unknown adjustable parameter of ANNs Designed parameters of MV-ANN are optimized using the optimization techniques of GA integrated with SQP.

Genetic Algorithm (GA)

Genetic algorithm is a global optimization procedure. Just like in nature, GAs work by selecting the best solutions from a group, combining them, and making small changes to create new solutions. Genetic algorithm is a search algorithm that uses a solution space to find the optimal solution to a problem. Its defining feature lies in the method it employs to conduct the search. The algorithm generates a "population" of potential solutions to a problem, allowing them to evolve over generations to discover improved solutions [73]. Genetic algorithms (GAs) are widely applicable for solving nonlinear differential equation and optimization method is inspired by natural selection, enabling them to explore complex solution spaces and solve various optimization issues in various fields. The study explores the use of a genetic algorithm for solving Two-Species Competition Model and comparing its efficiency with numerical methods. In 1970 John H. Holland presented the first practical use of GA [74]. GA has been employed to solve nonlinear differential equations [75, 76]. The genetic algorithm operates in a step-by-step manner. During the first step a random initial population is created. The population represents the algorithm's candidate solutions being evaluated throughout the algorithm's process, with new members added and others eliminated over generations. An individual is a single solution in the population. Individual fitness measures the effectiveness of their solution, with higher fitness values indicating better solutions, which are determined by the specific problem being addressed. Genetic algorithms aim to optimize problems by iteratively reaching their final value based on stopping criteria. The genetic algorithm operates in a step wise manner. GA continue with the following steps until the algorithm's termination conditions are satisfied.

1. Step 1: Randomization

   The initial stage generates a random initial population.

2. Step 2: Fitness Function

   The fitness function evaluates the alignment of ANN-predicted solutions with numerical solutions in the second GA phase, helping in the selection of optimal optimization settings.

3. Step 3: Reproduction

   The algorithm generates a series of new populations, using the current population to create the next in a recursive manner. Choose two individual at random from the population, making sure that those that are fitter have a better chance of getting chosen.
   Perform **crossover** on the two individuals to generate two new individuals.
   Let each individual of the new-population a random chance of **mutation**.
   Replace population with new-population for beter solution. This procedure is repeated until a minimum error is reached.

4. Step 4: Stopping Criteria

Genetic algorithms aim to optimize a problem by completing an iterative process under stopping conditions to reach its final value.

5. Step 5: As long as the ending condition is not yet met, proceed to step 2.

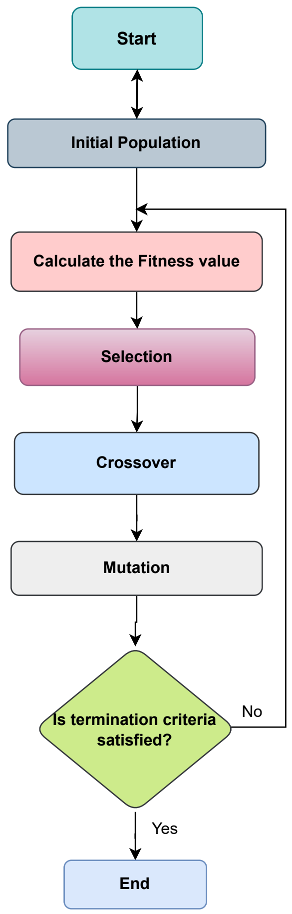

Figure 4.3: Genetic Algorithm Flowchart

Sequential Quadratic Programming (SQP)

Wilson's 1963 PhD thesis introduced the first SQP method to solve constrained nonlinear optimization problems. SQP is a local optimization algorithm. SQP focuses on solving problems by decomposing them down into smaller, manageable parts (QP sub problems). SQP methods approximate a solution by solving a series of QP subproblems, where a quadratic model of the objective function is minimized under linearized constraints. The subproblem in SQP method is solved through iteration using a QP method. One of the essential applications of SQP is its great accuracy in locating optimal solutions. SQP emphasis on quadratic sub problems enables it to rapidly converge to a solution, making it well-suited for complicated optimization tasks [77]. After Wilson's 1963 publication, there has been a significant transformation in SQP techniques. Since the late 1970s, Sequential Quadratic Programming (SQP) has emerged as the most effective method for solving nonlinear constrained optimization problems [78]. Numerous writers have extensively studied sequential quadratic programming (SQP) techniques during past decades [79, 80, 81, 82].

The performance of GA is improved by integrating it with SQP, a local optimization algorithm. SQP focuses on solving problems by decomposing them down into smaller, manageable parts (quadratic programming sub problems). One of the essential applications of SQP is its great accuracy in locating optimal solutions. SQP emphasis on quadratic sub problems enables it to rapidly converge to a solution, making it well-suited for complicated optimization tasks. Some of the applications of SQP are optimal control in direct numerical modeling of turbulent flow [83], Optimization of a multi-product economic production quantity problem [84]. MW-ANN designed parameters are optimized using GA and SQP algorithms to solve two species competition, a nonlinear problem. This combination is particularly useful for solving two species competition model, which is a complex mathematical problem. Using a software tool called MATLAB, which has built-in functions to help with optimization. The specific functions they are using are called 'GA' for Genetic Algorithms and 'FMINCON' for Sequential Quadratic Programming. MATLAB is like a tool box that can help in solving different type of problem making it easier to implement complex algorithms.

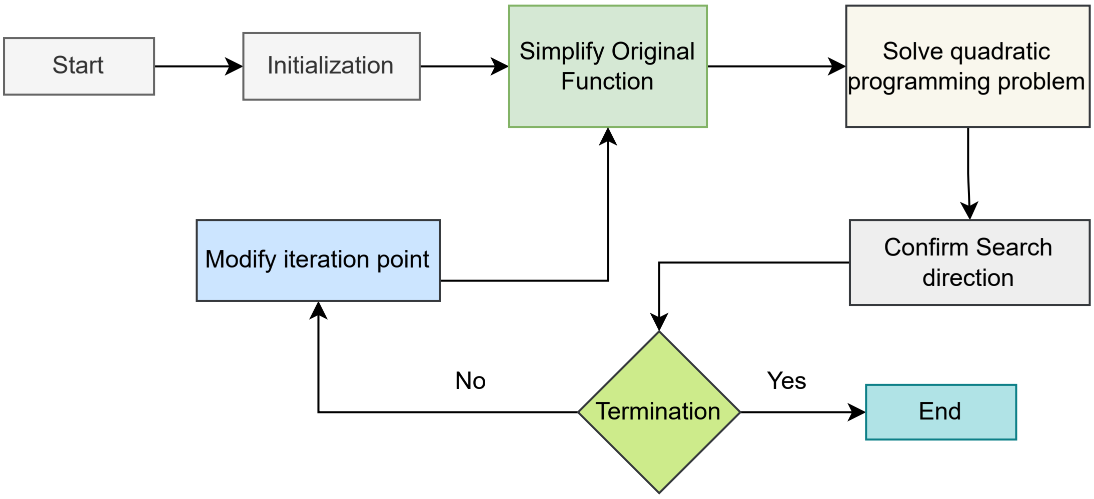

Figure 4.4: SQP flowchart

Hybrid GA-SQP

SQP requires fewer objective and constraint function evaluations compared to GA. As a deterministic algorithm, it can produce extremely highly accurate optimal results. SQP's local optima and noise sensitivity make it susceptible to constraints, while GA's global search approach increases the likelihood of identifying the global optimum. The GA should be utilized for a preliminary global search, with its outcomes serving as a guide for the subsequent local search. The combination of GA and SQP enhances a GA global search capacity and SQP local search precision. The GA stopping criteria, such as low generation, low population, or high tolerance, are established to cause the GA to stop early, The GA is supposed to find its best outcomes close to the actual global optimum.The SQP algorithm utilizes GA findings to identify its local optimum, which is the globally searched optimum, after conducting a local search [85].
This process of optimization through Hybrid GA-SQP consist of following steps

1. Step 1: Define the Objective Function

   The objective function is defined at the beginning of the optimization process, assessing the model's fitness or error using the provided parameters.

2. Step 2: Parameters for Genetic Algorithm (GA)

   The Genetic Algorithm (GA) is utilized as the primary global optimization method to determine the optimal set of parameters. The Genetic Algorithm (GA) is designed for 50 generations with a 100-person population size. It ended early if no improvement is seen for 40 generations, and a function tolerance was used for convergence. All 60 variables were kept within predetermined ranges for meaningful values.

3. Step 3: Run Genetic Algorithm

   The objective function is optimized by using the GA function in MATLAB is utilized to optimize across the specified search space, with the best solution obtained from GA ($WBest-GA$) optimal solution saved for future improvement.

4. Step 4: Parameters for Sequential Quadratic Programming (SQP)

   SQP was utilized as a local optimization technique to enhance the GA solution. This study utilized Sequential Quadratic Programming (SQP) due to its proven effectiveness in resolving constrained optimization issues. The algorithm, with a step tolerance of $1e-20$ and function tolerance of $1e-18$, can operate for 1500 iterations and ensures reliability and comprehensive evaluation.

5. Step 5: Hybrid GA-SQP Process

   The first guess for SQP was based on the best GA solution ($WBest-GA$), with 50 iterations for refinement and actions taken every cycle.

   The first guess for SQP was based on the best GA solution ($WBest-GA$), with 50 iterations for refinement and actions taken every cycle. The GA-derived parameters were optimized using SQP refinement using $fmincon$, with iterations, fitness, optimized values, and function evaluations documented.

6. Step 6: Save Results

   After optimization, all outcomes, including fitness values and optimized parameters, were saved in a MATLAB data file for future examination. The grapgical representation of hybrid GA-SQP is given in Fig.4.5.

4.5.5 Performance Measures

To evaluate the effectiveness of ANN-GA-SQP technique we will compute the following:

$$\text{Absolute Error} = \sum_{i=1}^{n} \left(|x_i - \hat{x}_i|, |y_i - \hat{y}_i|\right), \tag{4.16}$$

$$\text{Mean Square Error} = \frac{1}{n}\sum_{i=1}^{n} \left((x_i - \hat{x}_i)^2, (y_i - \hat{y}_i)^2\right), \tag{4.17}$$

$$\text{Mean Absolute Deviation} = \frac{1}{n}\sum_{i=1}^{n}\left(|x_i - \hat{x}_i|, |y_i - \hat{y}_i|\right). \tag{4.18}$$

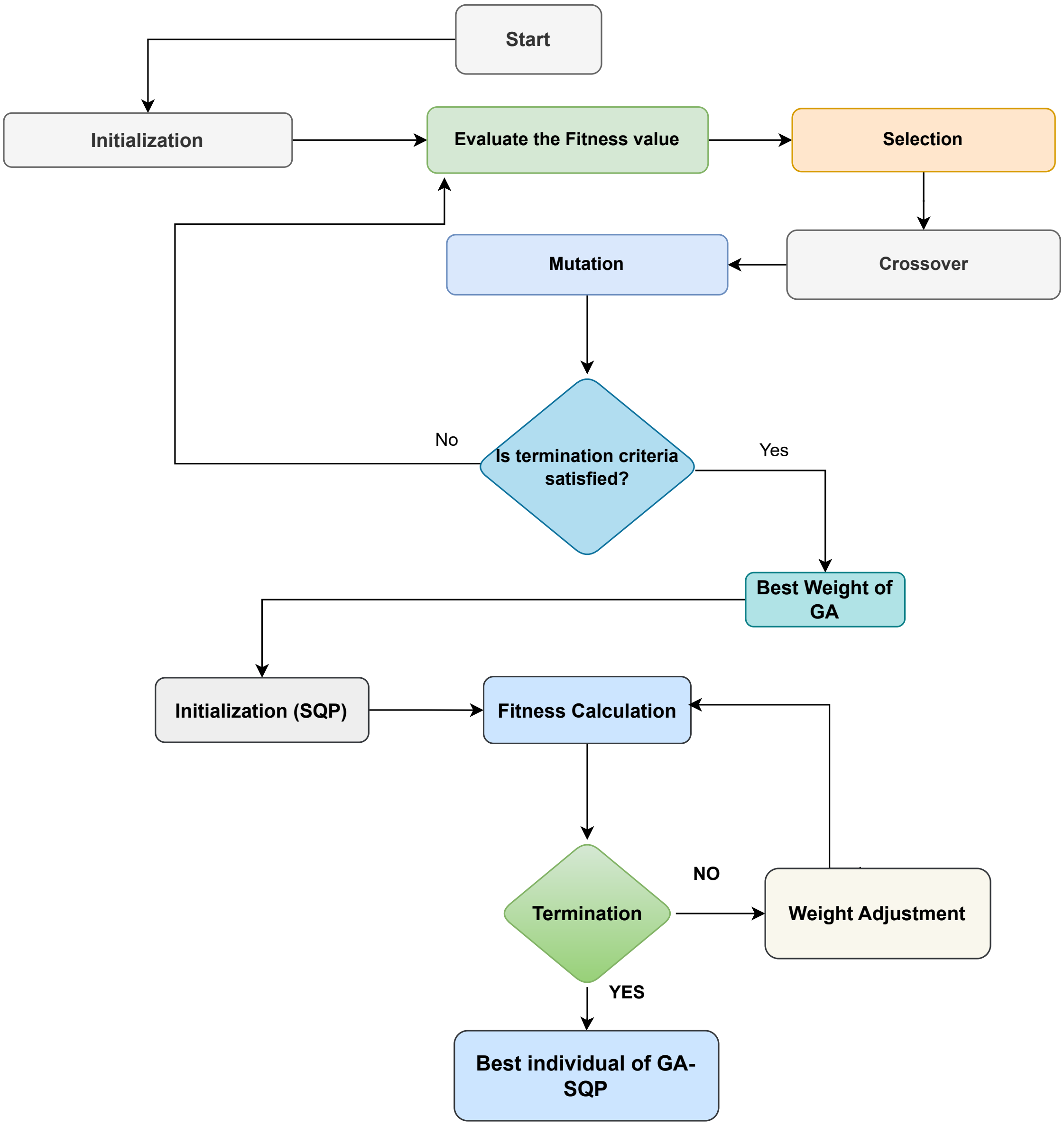

Figure 4.5: Hybrid GA-SQP Flowchart

4.6 Results and Discussion

This study utilized an Artificial Neural Network (ANN) architecture to numerically solve the Two-Species Competition Model. The ANN approximated the solution by reducing the error between the predicted and the exact solutions after the model equations were structured into an unsupervised learning problem. MATLAB was utilized for the development of the ANN due to its computational capabilities and integrated tools for neural network modeling and optimization. The Artifcial Neural Network design consisted of three layers, a single input layer, a hidden layer with Morlet wavelet activation functions, and an output layer representing the dependent variables $x(t)$ and $y(t)$. This study used an hybrid optimization technique that combined GA and SQP.

The ANN based numerical solution for the nonlinear coupled mathematical model of the Two-Species Competition Model is developed using appropriate parameter values is as follows

$$\frac{d\hat{x}}{dt} = 0.75\hat{x}\left(1 - \frac{\hat{x} + 1.2\hat{y}}{100}\right), \tag{4.19}$$

$$\frac{d\hat{y}}{dt} = 0.75\hat{y}\left(1 - \frac{\hat{y} + 0.8\hat{x}}{80}\right), \tag{4.20}$$

$$x(0) = 10, \tag{4.21}$$

$$y(0) = 20, \tag{4.22}$$

Parameter	**Values**
α	1.2
β	0.8
r_1	0.75
r_2	0.75
K_1	100
K_2	80

Table 4.1: List of parameters for Two-Species Competition Model

The Fitness Function becomes

$$\varepsilon_f = \left[\frac{d\hat{x}}{dt} - 0.75\hat{x}\left(1 - \frac{\hat{x} + 1.2\hat{y}}{100}\right)\right]^2 + \left[\frac{d\hat{y}}{dt} - 0.75\hat{y}\left(1 - \frac{\hat{y} + 0.8\hat{x}}{80}\right)\right]^2 + (\hat{x}(0) - 10)^2 + (\hat{y}(0) - 20)^2. \tag{4.23}$$

For the numerical solution of the nonlinear mathematical system of Two-Species Competition Model, the fitness function has been optimized through a hybrid computing procedure i.e. GA-SQP. The study utilized 50 independent runs with a 10 neuron network and Morlet wavelet activation functions in the hidden layer, resulting in 30 weights for each output variable, and presented the optimal weights obtained using a hybrid GA-SQP approach. The best weights are presented in Fig. 4.6 through ANN-GA-SQP.

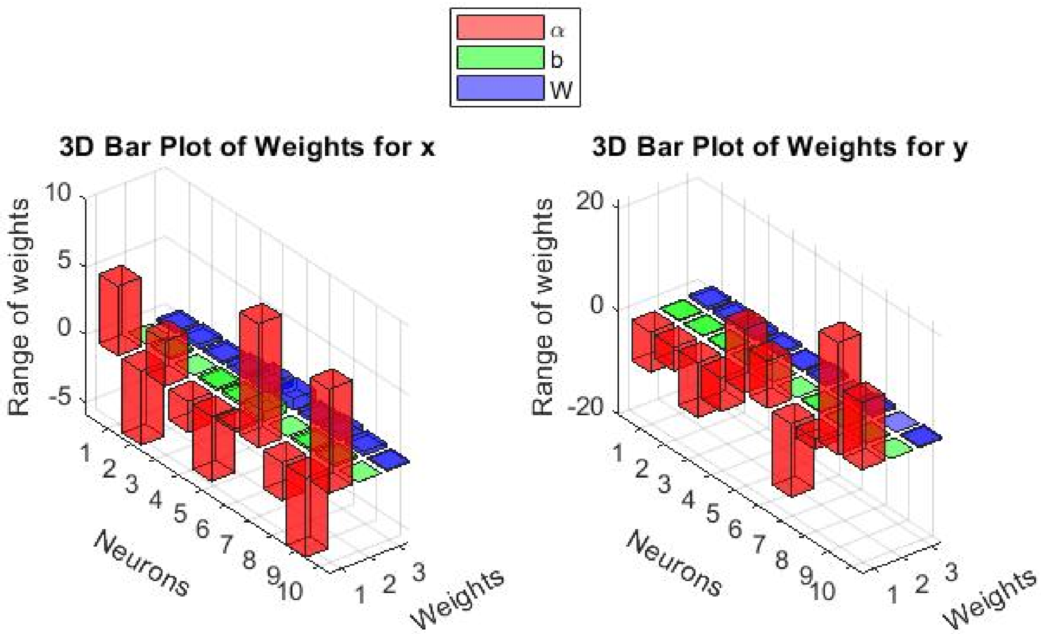

Figure 4.6: Best weights of the ANNs-GA-SQP

The study compares the results of RK numerical solvers and the proposed stochastic computing procedure MW-ANN-GA-SQP. Tables 4.2 and 4.3 provide a detailed comparison between the results produced by the proposed ANN model and those calculated using the numerical technique. The close similarity between the two sets of values highlights the ANN's strength in accurately approximating the solution. This consistency in results reflects the robustness of the ANN-based approach. The overlapping outcomes further suggest that the model performs reliably. In summary, the table reinforces the validity of the ANN method in capturing the dynamics of the numerical solution.

Table 4.2: Comparison of ANN and Numerical Solutions for $x(t)$

t	x_{ANN}	$x_{\text{Numerical}}$	Error
0	10.0000016	10	1.60E-06
0.1	10.50084889	10.50084834	5.50E-07
0.2	11.0127194	11.0127219	2.50E-06
0.3	11.5345326	11.53453797	5.37E-06
0.4	12.06511423	12.06511995	5.72E-06
0.5	12.60320383	12.6032072	3.37E-06
0.6	13.14746618	13.14746679	6.10E-07
0.7	13.69650593	13.69650671	7.80E-07
0.8	14.24888446	14.2488905	6.04E-06
0.9	14.80313806	14.80315278	1.47E-05
1	15.35779669	15.3578154	1.87E-05

Table 4.3: Comparison of ANN and Numerical Solutions for $y(t)$

t	y_{ANN}	$y_{\text{Numerical}}$	Error
0	20.00000066	20	6.60E-07
0.1	20.98552424	20.98556488	4.06E-05
0.2	21.99087475	21.99079943	7.53E-05
0.3	23.01357491	23.01336368	2.11E-04
0.4	24.0510246	24.05073294	2.92E-04
0.5	25.10051271	25.10021986	2.93E-04
0.6	26.15922993	26.15899997	2.30E-04
0.7	27.22428243	27.22414042	1.42E-04
0.8	28.29270609	28.29263123	7.49E-05
0.9	29.36148145	29.36141837	6.31E-05
1	30.42754906	30.42743795	1.11E-04

Figs.4.7 and 4.8 presents a comparison between the optimal solutions achieved using the ANNs-GA-SQP method for the Two-Species Competition Model and the corresponding reference solutions. The best results from the ANNs-GA-SQP are shown to overlap with the reference

solutions, demonstrating the accuracy of the suggested numerical approach.

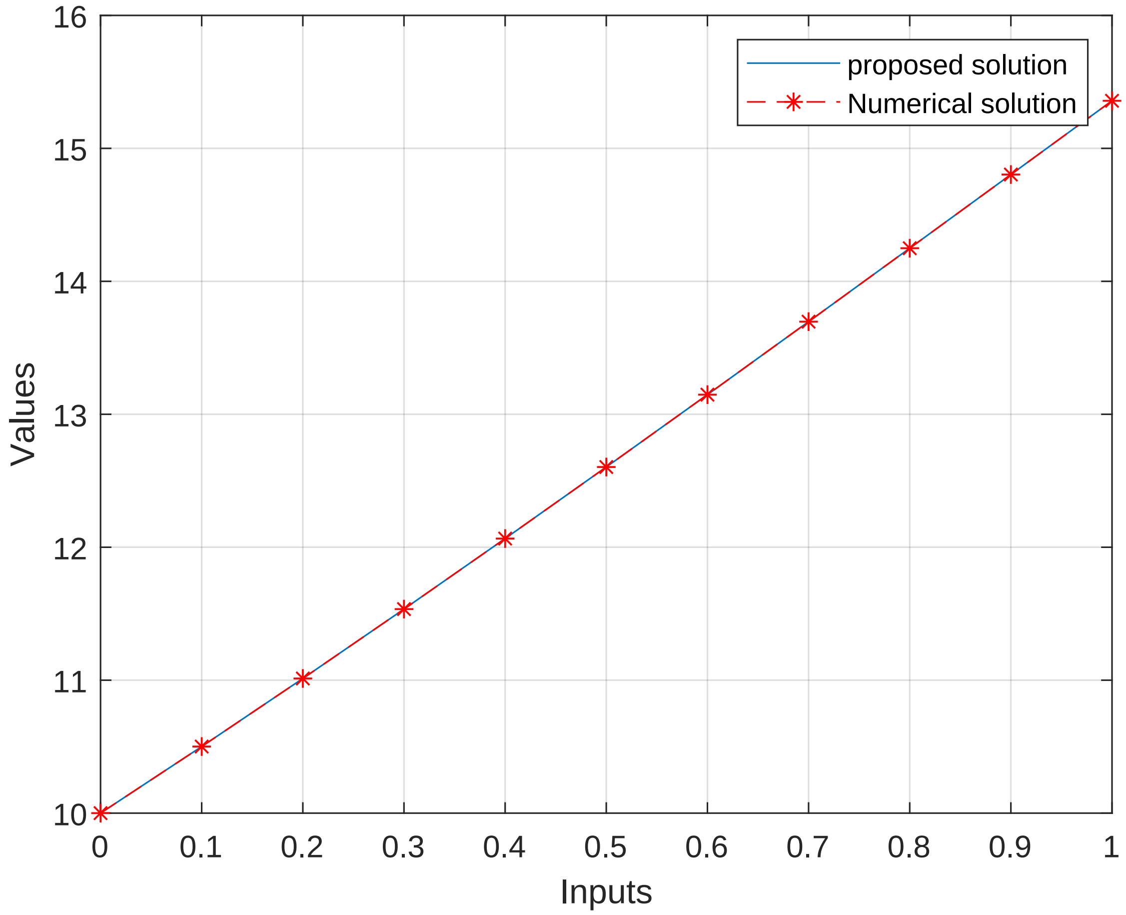

Figure 4.7: Result Comparison of x-species

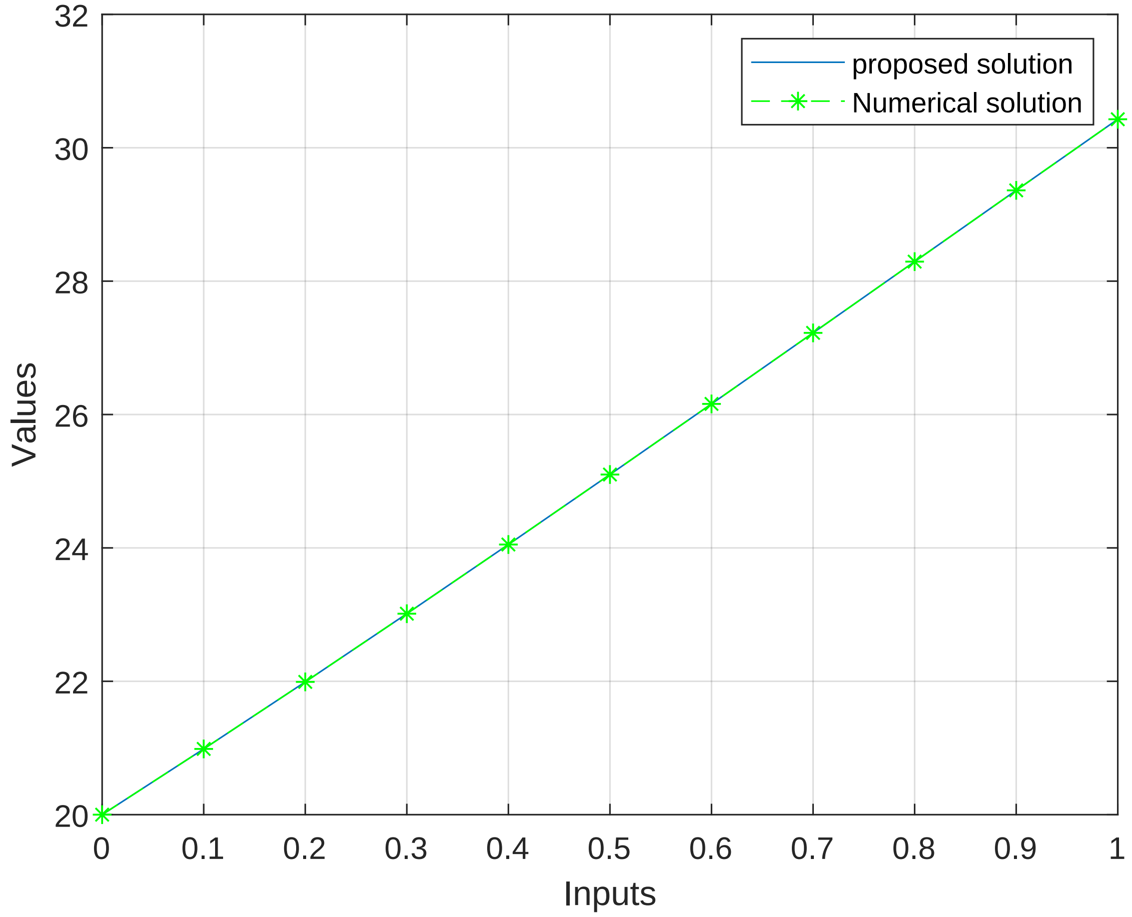

Figure 4.8: Result Comparison of y-species

The proposed scheme's precision is verified by performing absolute error values through the mean and best. The absolute error graph illustrates the difference at each point between the solutions estimated by the ANN model and the corresponding numerical results for each species in competition model. The absolute error values for Two-Species Competition Model are depicted in Fig. 4.10 which displays the absolute errors for species x and y between ANN-predicted solutions and reference numerical solutions like Runge-Kutta over a time interval $t \in [0, 1]$.

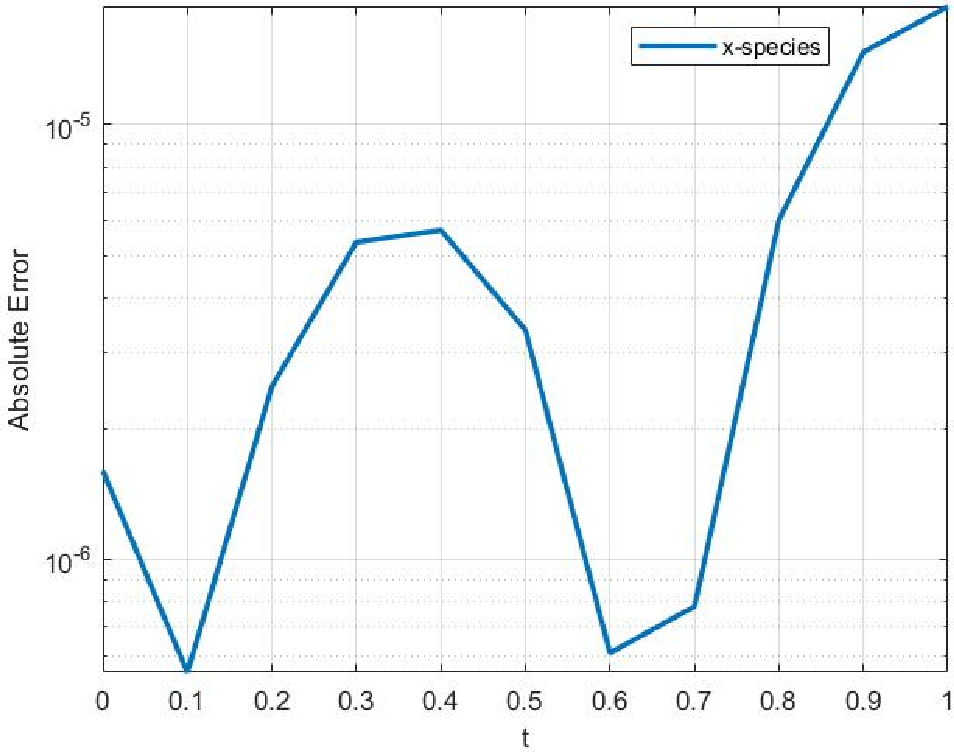

Figure 4.9: Absolute Error for $x-$species

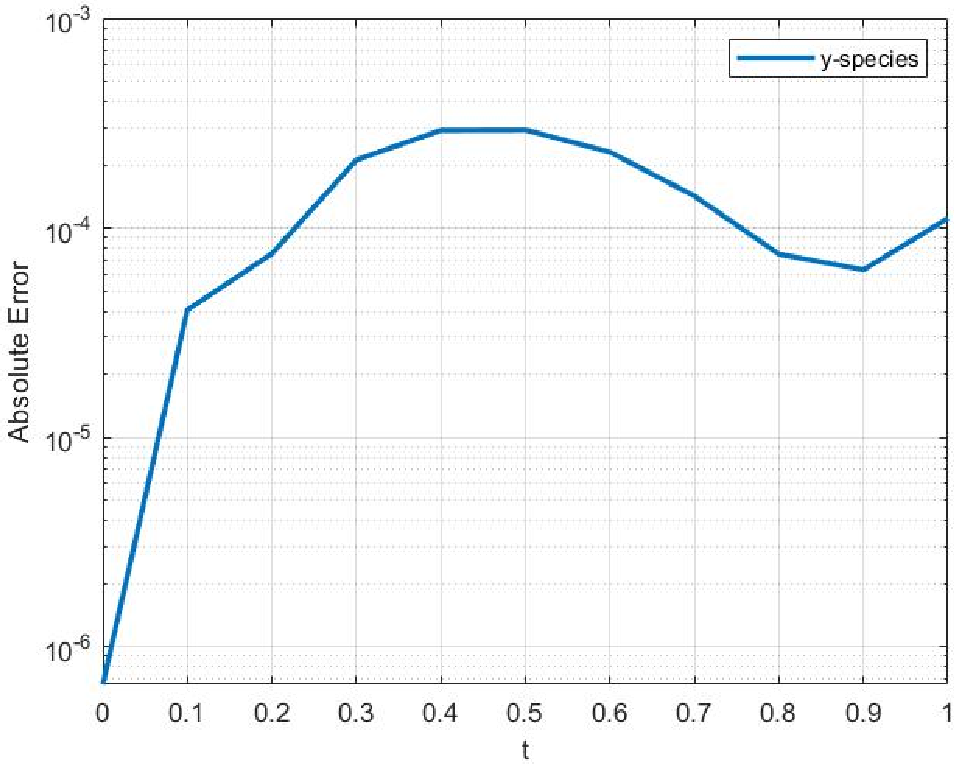

Figure 4.10: Absolute Error for $y-$species

It is evident that the ANN and numerical technique have similar results due to the extremely low absolute error for the x species over a wide timeframe, primarily in the 10^{-5} to 10^{-7} range, while the error for the y species ranges from 10^{-4} to 10^{-7} which is manageably minimal. Overall, the absolute error analysis confirms that the neural network successfully approximates the true solution with high precision, effectively capturing the non-linear interactions in the Two-Species Competition Model.

The performance and consistency of the proposed NN model were evaluated through the calculation of the Mean Square Error for 50 simulation runs.

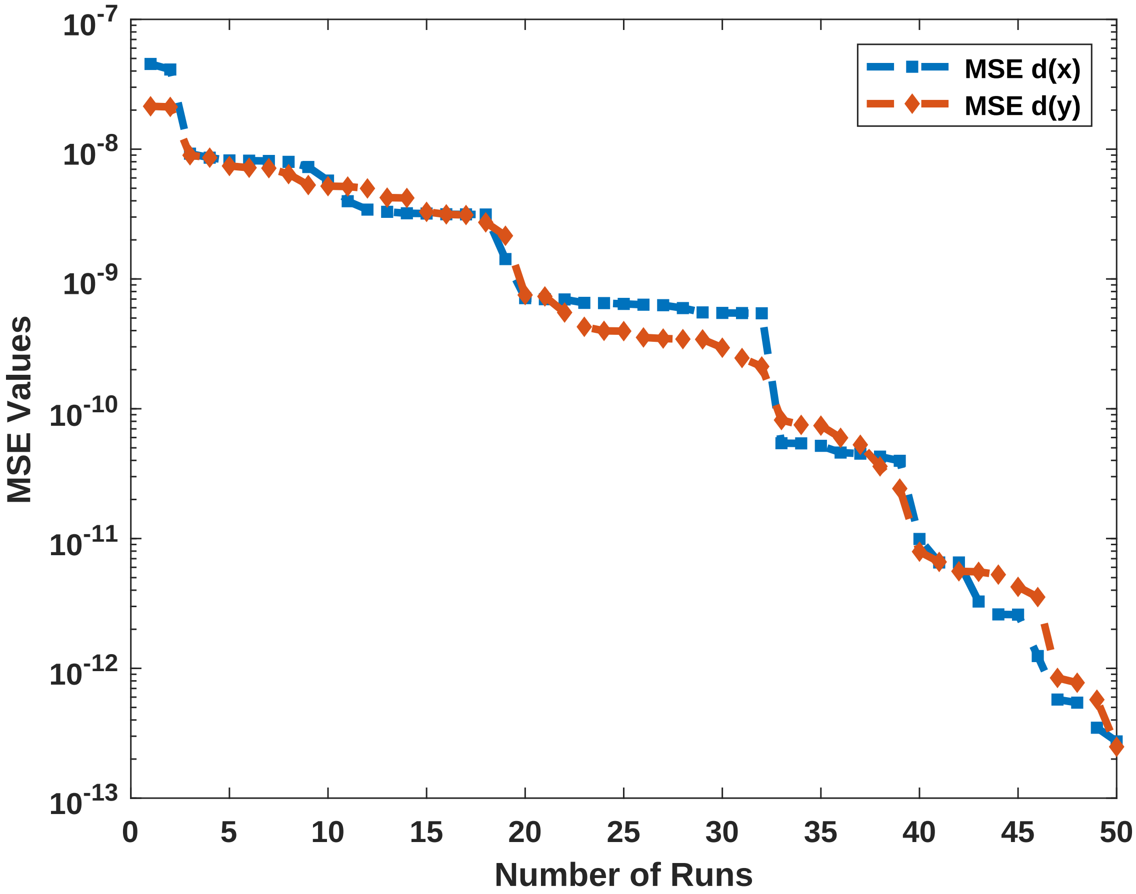

Figure 4.11: Mean Square Error

The accuracy and reliability of the Morlet wavelet-based ANN was evaluated by computing the mean square error for both species over 50 optimization runs. The graph's steady drop in MSE values for $\frac{dx}{dt}$ and $\frac{dy}{dt}$ illustrates the strong approximation capabilities of the model. The Morlet wavelet-based ANN optimized with the hybrid GA-SQP algorithm exhibits high accuracy, as indicated by the MSE values ranging from 10^{-7} to 10^{-13}. The results indicate that the neural network consistently produces very low MSE values for both species. The error associated with the x-component is the smallest, nearing 10^{-13}, while the y-component follows closely with slightly higher, yet still minimal, error values. This suggests that the model successfully captures the dynamics of both species with a high degree of accuracy.The model's stability and effectiveness in resolving the nonlinear Two-Species Competition System are demonstrated by its exceptionally low error values.

The Mean Absolute Deviation (MAD) is calculated from 50 simulation runs to assess the model's stability and accuracy. The consistent MAD values confirm the reliability of the Morlet wavelet-based neural network optimized through the hybrid GA-SQP method.

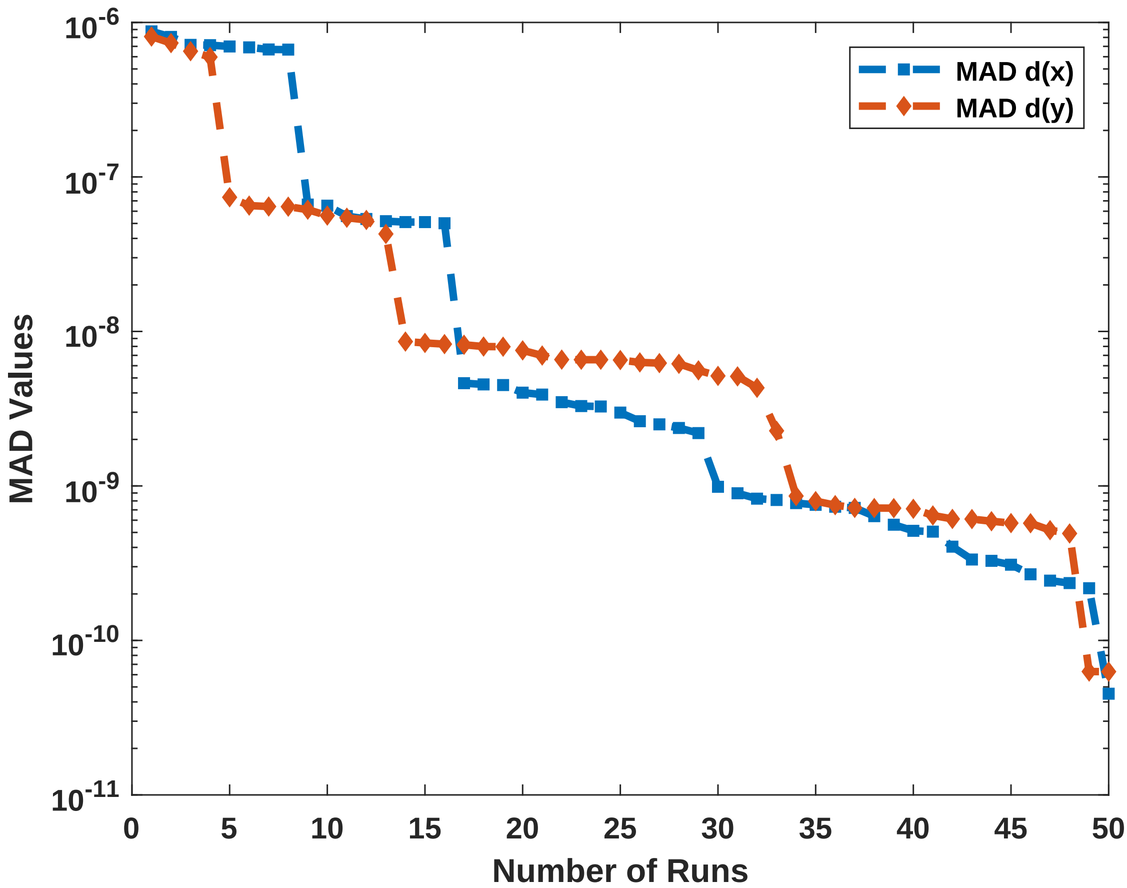

Figure 4.12: MAD Performance

MAD values are calculated using 4.18 for the morlet wavelet ANN technique.The graph illustrate the MAD values for the $x(t)$ and $y(t)$ components of the Two-Species Competition

Model over 50 optimization runs. The MAD evaluation across 50 optimization runs reveals that the error values for MAD $d(x)$ and MAD $d(y)$ both fall within the extremely accurate range of 10^{-6} to 10^{-11}. The network effectively simulates the dynamics of the first species, with MAD $d(x)$ reaches the smallest deviation among two species approaching values close to 10^{-11}. Similar to this MAD $d(y)$ maintains a low error range between 10^{-8} to 10^{-7}, indicating a well-captured behavior of the second species, with a slightly more complex nature. The Morlet wavelet-based ANN, using a hybrid GA-SQP technique, exhibits exceptional approximation capability due to its constant low deviation. These very small MAD values show that the model is highly reliable in capturing the complex interactions between the two species. They also confirm that the hybrid GA-SQP optimization method works well to reduce errors without causing overfitting.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The thesis presents a precise, biologically-inspired neuro-heuristic framework for solving nonlinear differential systems, primarily focusing on the Two-Species Competition Model. The model consists of a system of coupled nonlinear ODE that is commonly employed to represent population interactions between different species in ecological studies. The proposed method integrates ANN with deterministic and evolutionary optimization techniques to efficiently solve nonlinear systems with accuracy and stability. The proposed model is designed using an ANN that utilizes the Morlet wavelet as its activation function. Using a wavelet-based activation function improves the network's capability to identify complex and localized nonlinear patterns within the system, which makes it especially effective for modeling ecological interactions. The neural network's weights and biases were optimized using a hybrid approach that combined the local refinement power of SQP with the global search capability of GA. This hybrid GA-SQP technique ensures convergence towards globally optimal solutions by improving accuracy.

The performance of the model was thoroughly verified through multiple independent optimization runs (a total of 50). The accuracy of neural network solutions was evaluated using Mean Square Error (MSE) and Mean Absolute Deviation (MAD) as performance measures. For both species, the MAD values remained between 10^{-6} and 10^{-11}, while the MSE values continuously ranged from 10^{-7} to 10^{-13}. The consistently low error values highlight the high accuracy of the proposed method and also reflect its stability over multiple runs, indicating that the neural network can generalize well and remains reliable despite different initial conditions.

In addition, the performance of the proposed ANN model was compared with traditional numerical methods, including the fourth-order Runge-Kutta (RK4) and the Taylor series approach. The analysis showed that the ANN model, optimized through the hybrid GA-SQP method, achieved accuracy that was comparable to or even better than traditional numerical methods, particularly in maintaining stability over extended simulation periods and in the presence of nonlinear parameter changes.

To conclude, the GA-SQP optimized ANN using the Morlet wavelet offers a strong and adaptable method for numerically solving nonlinear DE like the Two-Species Competition Model. The proposed method has shown excellent accuracy, consistent convergence behavior, and reliable results across numerous test runs. Due to its strong performance, this framework has the potential to be extended to a wide range of nonlinear differential systems, particularly in cases where conventional numerical approaches prove inadequate. In future research, this strategy could be adapted for more complex models involving multiple species, systems with time delays, or ecological scenarios driven by real-world data.

5.2 Future Work

The study demonstrates the effectiveness of ANN in solving the Two-Species Competition Model, particularly when combined with hybrid GA-SQP optimization and Morlet wavelet activation function. The current model could be expanded to handle complex ecological interactions involving multiple species, providing broader ecological insights. The Morlet wavelet's effectiveness is being evaluated against alternative wavelet-based to find more ideal setups for specific differential equations, despite its effectiveness. The current optimization method GA-SQP could be enhanced by incorporating metaheuristic algorithms like Particle Swarm Optimization and hybrid swarm-intelligence-based strategies like PSO-ASA. The suggested framework may develop into a more complete and flexible system for modeling and resolving a variety of nonlinear differential equation issues in ecology and other fields as a result of these additions.

BIBLIOGRAPHY

[1] Z. Sabir, M. R. Ali, M. A. Z. Raja, and R. Sadat, "An efficient computational procedure to solve the biological nonlinear leptospirosis model using the genetic algorithms," *Soft Computing*, vol. 28, no. 1, pp. 607–625, 2024.

[2] M. Bilal, M. A. Z. Raja, I. Ahmad, R. Khan, and M. Shoaib, "Dynamical analysis of nonlinear combined drug therapy model for hiv infection: Bayesian regularization technique intelligent networks," *Biomedical Signal Processing and Control*, vol. 88, p. 105629, 2024.

[3] Z. Shah, M. A. Z. Raja, and M. Shoaib, "Design of bayesian stochastic networks for numerical treatment of williamson fluid stretching flow model with mixed convected heat generation," *Numerical heat transfer, part b: fundamentals*, vol. 86, no. 7, pp. 2030–2053, 2025.

[4] Z. Sabir, M. A. Z. Raja, H. A. Wahab, M. Shoaib, and J. G. Aguilar, "Integrated neuro-evolution heuristic with sequential quadratic programming for second-order prediction differential models," *Numerical Methods for Partial Differential Equations*, vol. 40, no. 1, p. e22692, 2024.

[5] A. Shloof, N. Senu, A. Ahmadian, M. Pakdaman, and S. Salahshour, "A new iterative technique for solving fractal-fractional differential equations based on artificial neural network in the new generalized caputo sense," *Engineering with Computers*, vol. 39, no. 1, pp. 505–515, 2023.

[6] A. Ali, N. Senu, N. Wahi, N. Almakayeel, and A. Ahmadian, "An adaptive algorithm for numerically solving fractional partial differential equations using hermite wavelet artificial neural networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 137, p. 108121, 2024.

[7] M. K. Basit, T. A. Khan, J. I. Janjua, A. Hussain, H. Abdullah, and S. A. Ramay, “An efficient approach for solving second order or higher ordinary differential equations using ann,” *Journal of Computing & Biomedical Informatics*, vol. 5, no. 02, pp. 93–102, 2023.

[8] L. Yan, Z. Sabir, E. Ilhan, M. A. Z. Raja, W. Gao, and H. M. Baskonus, “Design of a computational heuristic to solve the nonlinear liénard differential model,” *Computer Modeling in Engineering & Sciences*, vol. 136, no. 1, pp. 201–221, 2023.

[9] M. Umar, Z. Sabir, M. A. Z. Raja, F. Amin, T. Saeed, and Y. G. Sanchez, “Design of intelligent computing solver with morlet wavelet neural networks for nonlinear predator–prey model,” *Applied Soft Computing*, vol. 134, p. 109975, 2023.

[10] C. Biswas, A. Singh, M. Chopra, and S. Das, “Study of fractional-order reaction-advection-diffusion equation using neural network method,” *Mathematics and Computers in Simulation*, vol. 208, pp. 15–27, 2023.

[11] M. Wu, J. Zhang, Z. Huang, X. Li, and Y. Dong, “Numerical solutions of wavelet neural networks for fractional differential equations,” *Mathematical Methods in the Applied Sciences*, vol. 46, no. 3, pp. 3031–3044, 2023.

[12] Z. Sabir, M. A. Zahoor Raja, J. L. Guirao, and T. Saeed, “Design of mayer wavelet neural networks for solving functional nonlinear singular differential equation,” *Mathematical Problems in Engineering*, vol. 2022, no. 1, p. 1213370, 2022.

[13] K. Günel and İ. Gör, “Solving dirichlet boundary problems for odes via swarm intelligence,” *Mathematical Sciences*, vol. 16, no. 4, pp. 325–341, 2022.

[14] Z. Sabir, H. A. Wahab, T. G. Nguyen, G. C. Altamirano, F. Erdoğan, and M. R. Ali, “Intelligent computing technique for solving singular multi-pantograph delay differential equation,” *Soft Computing*, vol. 26, no. 14, pp. 6701–6713, 2022.

[15] Z. Sabir, D. Baleanu, M. R. Ali, and R. Sadat, “A novel computing stochastic algorithm to solve the nonlinear singular periodic boundary value problems,” *International Journal of Computer Mathematics*, vol. 99, no. 10, pp. 2091–2104, 2022.

[16] N. Yadav, T. T. Ngo, and J. H. Kim, “An algorithm for numerical solution of differential equations using harmony search and neural networks,” *J. Appl. Anal. Comput*, vol. 12, pp. 1277–1293, 2022.

[17] M. Umar, Kusen, M. A. Z. Raja, Z. Sabir, and Q. Al-Mdallal, "A computational framework to solve the nonlinear dengue fever sir system," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 25, no. 16, pp. 1821–1834, 2022.

[18] Z. Sabir, M. A. Z. Raja, C. M. Khalique, and C. Unlu, "Neuro-evolution computing for nonlinear multi-singular system of third order emden–fowler equation," *Mathematics and Computers in Simulation*, vol. 185, pp. 799–812, 2021.

[19] Z. Sabir, M. A. Z. Raja, M. Umar, and M. Shoaib, "Neuro-swarm intelligent computing to solve the second-order singular functional differential model," *The European Physical Journal Plus*, vol. 135, no. 6, p. 474, 2020.

[20] M. Umar, Z. Sabir, and M. A. Z. Raja, "Intelligent computing for numerical treatment of nonlinear prey–predator models," *Applied Soft Computing*, vol. 80, pp. 506–524, 2019.

[21] J. Wildenberg, J. Vano, and J. Sprott, "Complex spatiotemporal dynamics in lotka–volterra ring systems," *ecological complexity*, vol. 3, no. 2, pp. 140–147, 2006.

[22] B. Batiha, "Comparison of numerical methods for solving one species lotka-volterra equation," *Analele Universit atii Oradea, Fasc. Matematica*, vol. 19, pp. 243–253, 2012.

[23] Y. Aksoy, Ü. Göktaş, M. Pakdemirli, and İ. T. Dolapçı, "Application of perturbation–iteration method to lotka–volterra equations," *Alexandria Engineering Journal*, vol. 55, no. 2, pp. 1661–1666, 2016.

[24] N. S. A. B. Razali and F. A. Abdullah, "Numerical methods for competitive hunters model," in *AIP Conference Proceedings*, vol. 1522, no. 1. American Institute of Physics, 2013, pp. 140–147.

[25] S. Paul, S. P. Mondal, P. Bhattacharya, and K. Chaudhuri, "Some comparison of solutions by different numerical techniques on mathematical biology problem," *International Journal of Differential Equations*, vol. 2016, no. 1, p. 8921710, 2016.

[26] A. Gupta and S. S. Ray, "Wavelet methods for solving fractional order differential equations," *Mathematical Problems in Engineering*, vol. 2014, no. 1, p. 140453, 2014.

[27] K. Nisar, Z. Sabir, M. A. Z. Raja, A. A. A. Ibrahim, F. Erdogan, M. R. Haque, J. J. Rodrigues, and D. B. Rawat, "Design of morlet wavelet neural network for solving a class of singular pantograph nonlinear differential models," *IEEE Access*, vol. 9, pp. 77 845–77 862, 2021.

[28] Z. Zainuddin and O. Pauline, "Modified wavelet neural network in function approximation and its application in prediction of time-series pollution data," *Applied Soft Computing*, vol. 11, no. 8, pp. 4866–4874, 2011.

[29] S. J. Farlow, *An Introduction to Differential Equations and their Applications*. Courier Corporation, 2006.

[30] E. L. Ince, *Ordinary Differential Equations*. Courier Corporation, 1956.

[31] W. A. Strauss, *Partial Differential Equations: An Introduction*. John Wiley & Sons, 2007.

[32] N. Yadav, A. Yadav, M. Kumar *et al.*, *An Introduction to Neural Network Methods for Differential Equations*. Springer, 2015, vol. 1.

[33] L. F. Shampine, J. Kierzenka, M. W. Reichelt *et al.*, "Solving boundary value problems for ordinary differential equations in matlab with bvp4c," *Tutorial notes*, vol. 2000, pp. 1–27, 2000.

[34] B. Biswas, S. Chatterjee, S. Mukherjee, and S. Pal, "A discussion on euler method: A review," *Electronic Journal of Mathematical Analysis and Applications*, vol. 1, no. 2, pp. 2090–2792, 2013.

[35] J. Butcher, "Runge–kutta methods for ordinary differential equations," *Numerical Analysis and Optimization: NAO-III, Muscat, Oman, January 2014*, pp. 37–58, 2015.

[36] G. Corliss and Y. Chang, "Solving ordinary differential equations using taylor series," *ACM Transactions on Mathematical Software (TOMS)*, vol. 8, no. 2, pp. 114–144, 1982.

[37] R. J. LeVeque, "Finite difference methods for differential equations," *Draft version for use in AMath*, vol. 585, no. 6, p. 112, 1998.

[38] A. Felaco, C. G. Opran, and G. Lamanna, "Numerical analysis techniques for studying the structural strength of a vehicle in terms of absorbed energy," in *Macromolecular Symposia*, vol. 413, no. 3. Wiley Online Library, 2024, p. 2400050.

[39] T. Weiland, "On the numerical solution of maxwell's equations and applications in the field of accelerator physics," *Part. Accel.*, vol. 15, no. DESY-84-006, pp. 245–292, 1984.

[40] K. Alhumaizi, R. Henda, and M. Soliman, "Numerical analysis of a reaction–diffusion–convection system," *Computers & chemical engineering*, vol. 27, no. 4, pp. 579–594, 2003.

[41] R. L. Burden and J. D. Faires, "Numerical analysis (nineth edition)," *Thomson Brooks/Cole*, pp. 57–58, 2010.

[42] P. Mohazzabi and J. L. Becker, "Numerical solution of differential equations by direct taylor expansion," *Journal of Applied Mathematics and Physics*, vol. 5, no. 3, pp. 623–630, 2017.

[43] M. Z. bin Mohamed, R. Hidayat, N. N. binti Suhaizi, M. K. H. bin Mahmud, S. N. binti Baharuddin *et al.*, "Artificial intelligence in mathematics education: A systematic literature review," *International Electronic Journal of Mathematics Education*, vol. 17, no. 3, p. em0694, 2022.

[44] J.-P. Haton, "A brief introduction to artificial intelligence," *IFAC Proceedings Volumes*, vol. 39, no. 4, pp. 8–16, 2006.

[45] J. Zupan, "Introduction to artificial neural network (ann) methods: what they are and how to use them," *Acta Chimica Slovenica*, vol. 41, no. 3, p. 327, 1994.

[46] G. Kutyniok, "The mathematics of artificial intelligence," in *Proceedings of the International Congress of Mathematicians*, 2022.

[47] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[48] S. Walczak, "Artificial neural networks," in *Encyclopedia of Information Science and Technology, Fourth Edition*. IGI Global Scientific Publishing, 2018, pp. 120–131.

[49] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[50] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2020, pp. 281–286.

[51] J.-h. Zhang and S.-f. Sun, "A novel chaotic neural network with morlet-wavelet activation function," in *2011 Second International Conference on Mechanic Automation and Control Engineering*. IEEE, 2011, pp. 29–32.

[52] J. Zupan, "Introduction to artificial neural network (ann) methods: what they are and how to use them," *Acta Chimica Slovenica*, vol. 41, no. 3, p. 327, 1994.

[53] M. H. Sazlı, "A brief review of feed-forward neural networks," *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006.

[54] T. D. Bui, D. K. Nguyen, and T. D. Ngo, "Supervising an unsupervised neural network," in *2009 First Asian Conference on Intelligent Information and Database Systems*. IEEE, 2009, pp. 307–312.

[55] R. Sharma, K. Sharma, and A. Khanna, "Study of supervised learning and unsupervised learning," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 6, pp. 588–593, 2020.

[56] M. R. Admon, N. Senu, A. Ahmadian, Z. A. Majid, and S. Salahshour, "A new efficient algorithm based on feedforward neural network for solving differential equations of fractional order," *Communications in Nonlinear Science and Numerical Simulation*, vol. 117, p. 106968, 2023.

[57] A. Shloof, N. Senu, A. Ahmadian, M. Pakdaman, and S. Salahshour, "A new iterative technique for solving fractal-fractional differential equations based on artificial neural network in the new generalized caputo sense," *Engineering with Computers*, vol. 39, no. 1, pp. 505–515, 2023.

[58] A. Ali, N. Senu, N. Wahi, N. Almakayeel, and A. Ahmadian, "An adaptive algorithm for numerically solving fractional partial differential equations using hermite wavelet artificial neural networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 137, p. 108121, 2024.

[59] S. Effati and M. Pakdaman, "Artificial neural network approach for solving fuzzy differential equations," *Information Sciences*, vol. 180, no. 8, pp. 1434–1457, 2010.

[60] Z. Sabir, H. A. Wahab, M. Umar, M. G. Sakar, and M. A. Z. Raja, "Novel design of morlet wavelet neural network for solving second order lane–emden equation," *Mathematics and Computers in Simulation*, vol. 172, pp. 1–14, 2020.

[61] M. A. Z. Raja, M. A. R. Khan, T. Mahmood, U. Farooq, and N. I. Chaudhary, "Design of bio-inspired computing technique for nanofluidics based on nonlinear jeffery–hamel flow equations," *Canadian Journal of Physics*, vol. 94, no. 5, pp. 474–489, 2016.

[62] A. J. Lotka, "The growth of mixed populations: two species competing for a common food supply," *Journal of the Washington academy of Sciences*, vol. 22, no. 16/17, pp. 461–469, 1932.

[63] M. G. Survey, *Maryland Geological Survey: Lower Cretaceous*. Johns Hopkins Press, 1911, vol. 4.

[64] A. J. Lotka, "The growth of mixed populations: two species competing for a common food supply," *Journal of the Washington academy of Sciences*, vol. 22, no. 16/17, pp. 461–469, 1932.

[65] L. Edelstein-Keshet, *Mathematical models in biology*. SIAM, 2005.

[66] V. Volterra, "Variations and fluctuations of the number of individuals in animal species living together," *ICES Journal of marine science*, vol. 3, no. 1, pp. 3–51, 1928.

[67] M. N. A. Manaf, N. F. Fauzi, N. S. A. Bakhtiar, N. I. Khairudin, and H. Z. A. Halim, "Comparative analysis of taylor series and runge-kutta fehlberg methods in solving the lotka-volterra competitive model," *Applied Mathematics and Computational Intelligence (AMCI)*, vol. 12, no. 3, pp. 91–103, 2023.

[68] M. R. Admon, N. Senu, A. Ahmadian, Z. A. Majid, and S. Salahshour, "A new efficient algorithm based on feedforward neural network for solving differential equations of fractional order," *Communications in Nonlinear Science and Numerical Simulation*, vol. 117, p. 106968, 2023.

[69] A. Ali, N. Senu, N. Wahi, N. Almakayeel, and A. Ahmadian, "An adaptive algorithm for numerically solving fractional partial differential equations using hermite wavelet artificial neural networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 137, p. 108121, 2024.

[70] S. Effati and M. Pakdaman, "Artificial neural network approach for solving fuzzy differential equations," *Information Sciences*, vol. 180, no. 8, pp. 1434–1457, 2010.

[71] A. Mehmood, A. Zameer, and M. A. Z. Raja, "Intelligent computing to analyze the dynamics of magnetohydrodynamic flow over stretchable rotating disk model," *Applied Soft Computing*, vol. 67, pp. 8–28, 2018.

[72] M. A. Z. Raja, F. H. Shah, E. S. Alaidarous, and M. I. Syam, "Design of bio-inspired heuristic technique integrated with interior-point algorithm to analyze the dynamics of heartbeat model," *Applied Soft Computing*, vol. 52, pp. 605–629, 2017.

[73] S. M. Thede, "An introduction to genetic algorithms," *Journal of Computing Sciences in Colleges*, vol. 20, no. 1, pp. 115–123, 2004.

[74] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM journal on computing*, vol. 2, no. 2, pp. 88–105, 1973.

[75] Z. Abo-Hammour, O. Abu Arqub, S. Momani, and N. Shawagfeh, "Optimization solution of troesch's and bratu's problems of ordinary type using novel continuous genetic algorithm," *Discrete Dynamics in Nature and Society*, vol. 2014, no. 1, p. 401696, 2014.

[76] O. A. Arqub and Z. Abo-Hammour, "Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm," *Information sciences*, vol. 279, pp. 396–415, 2014.

[77] E. Philip and W. Elizabeth, "Sequential quadratic programming methods," *UCSD Department of Mathematics Technical Report NA-10-03 August*, 2010.

[78] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica*, vol. 4, pp. 1–51, 1995.

[79] J. Cai, Q. Li, L. Li, H. Peng, and Y. Yang, "A hybrid fcaso-sqp method for solving the economic dispatch problems with valve-point effects," *Energy*, vol. 38, no. 1, pp. 346–353, 2012.

[80] C. Gu and D. Zhu, "A non-monotone line search multidimensional filter-sqp method for general nonlinear programming," *Numerical Algorithms*, vol. 56, pp. 537–559, 2011.

[81] J.-B. Jian, C.-M. Tang, Q.-J. Hu, and H.-Y. Zheng, "A new superlinearly convergent strongly subfeasible sequential quadratic programming algorithm for inequality-constrained optimization," *Numerical Functional Analysis and Optimization*, vol. 29, no. 3-4, pp. 376–409, 2008.

[82] L. Qi and Y.-F. Yang, "A globally and superlinearly convergent sqp algorithm for nonlinear constrained optimization," *Journal of Global Optimization*, vol. 21, pp. 157–184, 2001.

[83] H. Badreddine, S. Vandewalle, and J. Meyers, "Sequential quadratic programming (sqp) for optimal control in direct numerical simulation of turbulent flow," *Journal of Computational Physics*, vol. 256, pp. 1–16, 2014.

[84] S. H. R. Pasandideh, S. T. A. Niaki, and A. Gharaei, "Optimization of a multiproduct economic production quantity problem with stochastic constraints using sequential quadratic programming," *Knowledge-Based Systems*, vol. 84, pp. 98–107, 2015.

[85] A. Belloufi, M. Assas, and I. Rezgui, "Optimization of cutting conditions in multi-pass turning using hybrid genetic algorithm-sequential quadratic programming," *J Appl Mech Eng*, vol. 1, no. 1, pp. 3–7, 2012.